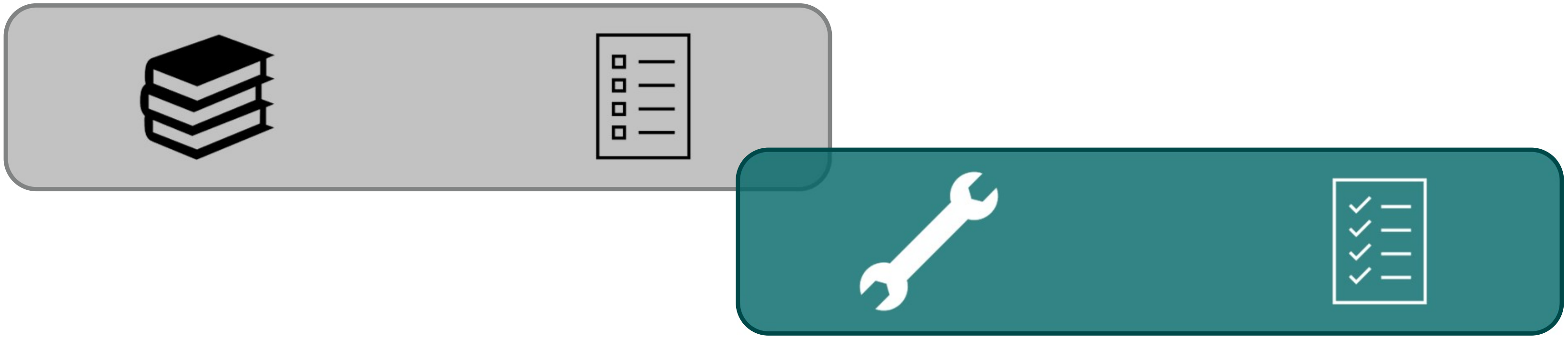


Reproducibility in Data Preprocessing: An Evaluation of Open Source Tools

Simon Grimm

Content

Introduction

Fundamentals

Related Work

Methods and Design

Results

Discussion

Summary and Outlook

Q&A

Introduction

- **Reproducibility** - vital and underpins trust in science
  - Ongoing and enhanced focus across different scientific domains and industries [Soh23]
  - Prevalence of data work as a major challenge [Fei+20]

- **Data Preprocessing** – make data suitable for analysis
  - foundation for data mining [AKV19], data science projects [ATSO17], data analysis [Fam+97], machine learning
  - impacts any derived conclusions, model quality, and model fairness [GZ19] [BR21]

- **Open Source** - „The bigger the problem, the more developers are drawn, like magnets, to work on it” [BCG21]
  - Integral to business
  - benefits reproducibility by fostering trust, enabling collaborative work, and emphasizing the value of software and data as artifacts for learning and sharing knowledge [Bar22]

Introduction

Motivation Purpose of this Work

Investigated research questions:

RQ1: What are the requirements for reproducible data preprocessing?

RQ2: What are promising open-source tools to enable reproducible data preprocessing?

RQ3: To what extent do existing open-source tools support reproducible data preprocessing?

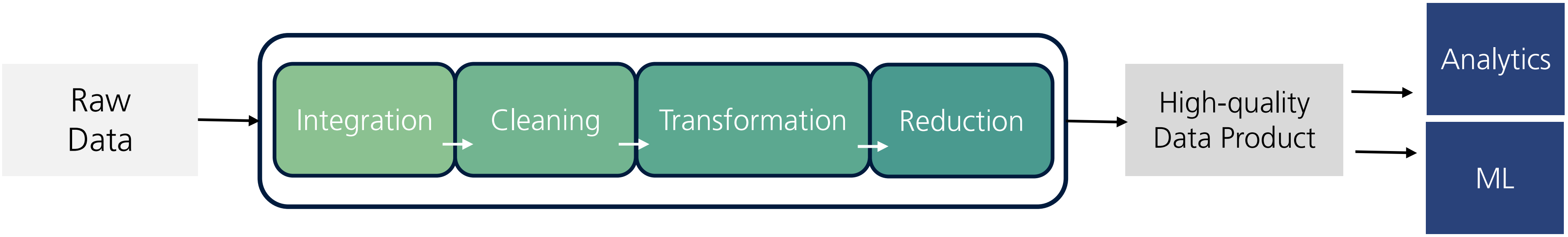
Fundamentals

- Non-conform standard across different scientific domains [GFI16]
- Often adapted according to a specific context [GK18]
- The Association for Computing Machinery proposes the following terminology[Noab]:
  - **Repeatability**: Same team, same experimental setup
    - *“a researcher can reliably repeat her own computation”*
  - **Reproducibility**: Different team, same experimental setup
    - *“an independent group can obtain the same result using the author’s own artifacts”*
  - **Replicability**: Different team, different experimental setup
    - *„independent group can obtain the same result using artifacts which they develop completely independently.* "

Fundamentals

Reproducibility Data Preprocessing

Data preprocessing comprises all necessary concepts and methods to transform raw data to a high-quality data product that satisfies the requirements for further usage [HKP12].

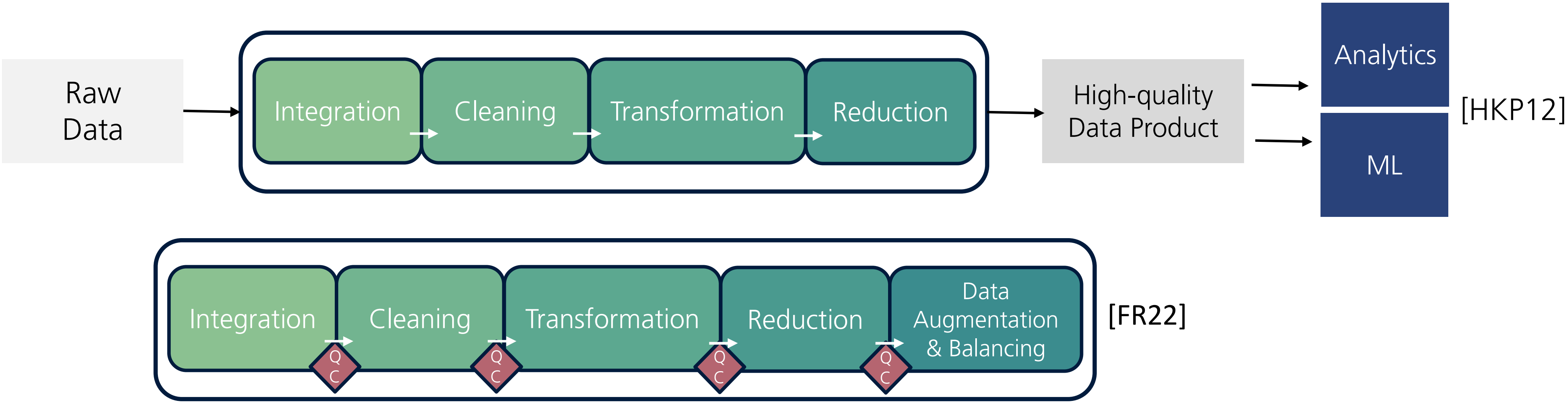


[HKP12]

Fundamentals

Reproducibility Data Preprocessing

Data preprocessing comprises all necessary concepts and methods to transform raw data to a high-quality data product that satisfies the requirements for further usage [HKP12].



Raw Data

Integration

Cleaning

Transformation

Reduction

High-quality Data Product

Analytics

ML

[HKP12]

Integration

Cleaning

Transformation

Reduction

Data Augmentation & Balancing

QC

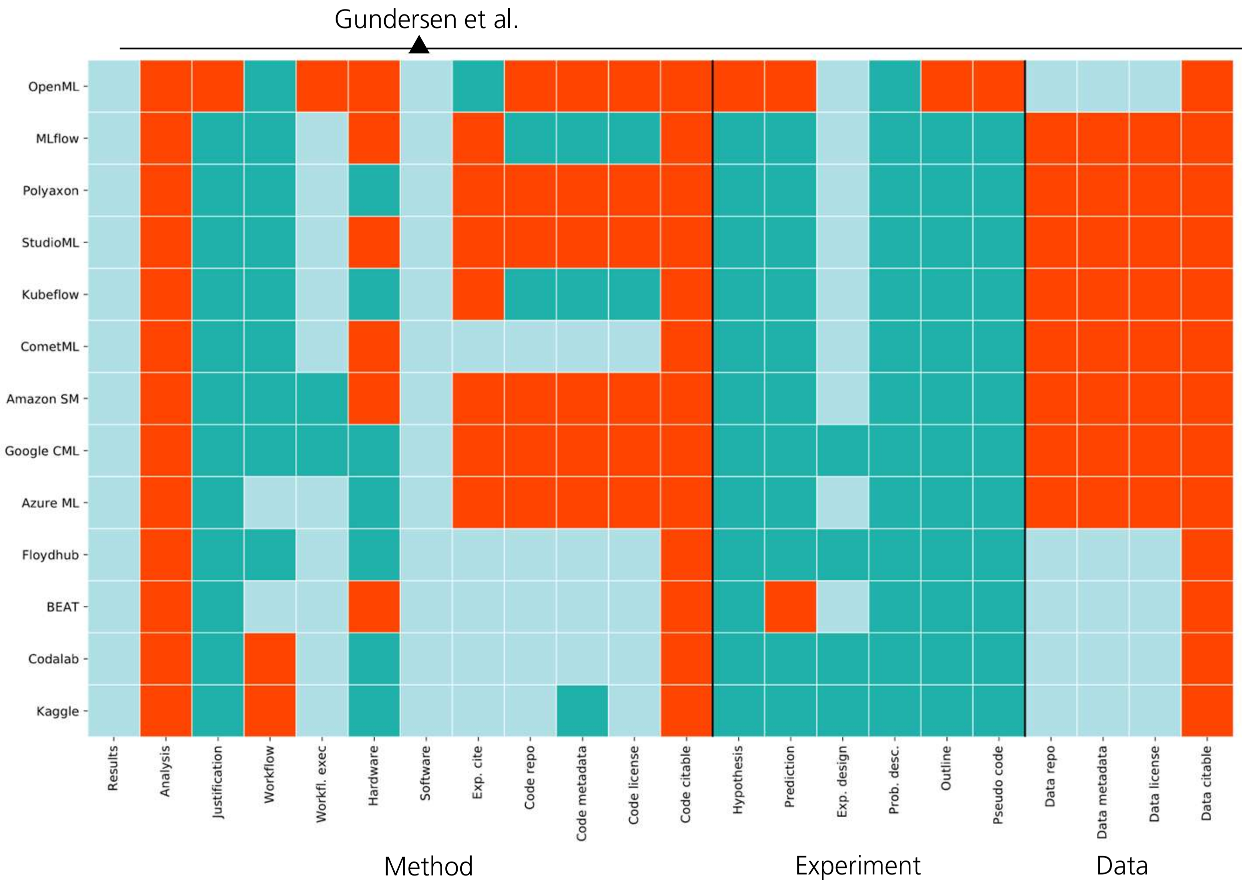
QC

QC

QC

[FR22]

Related Work



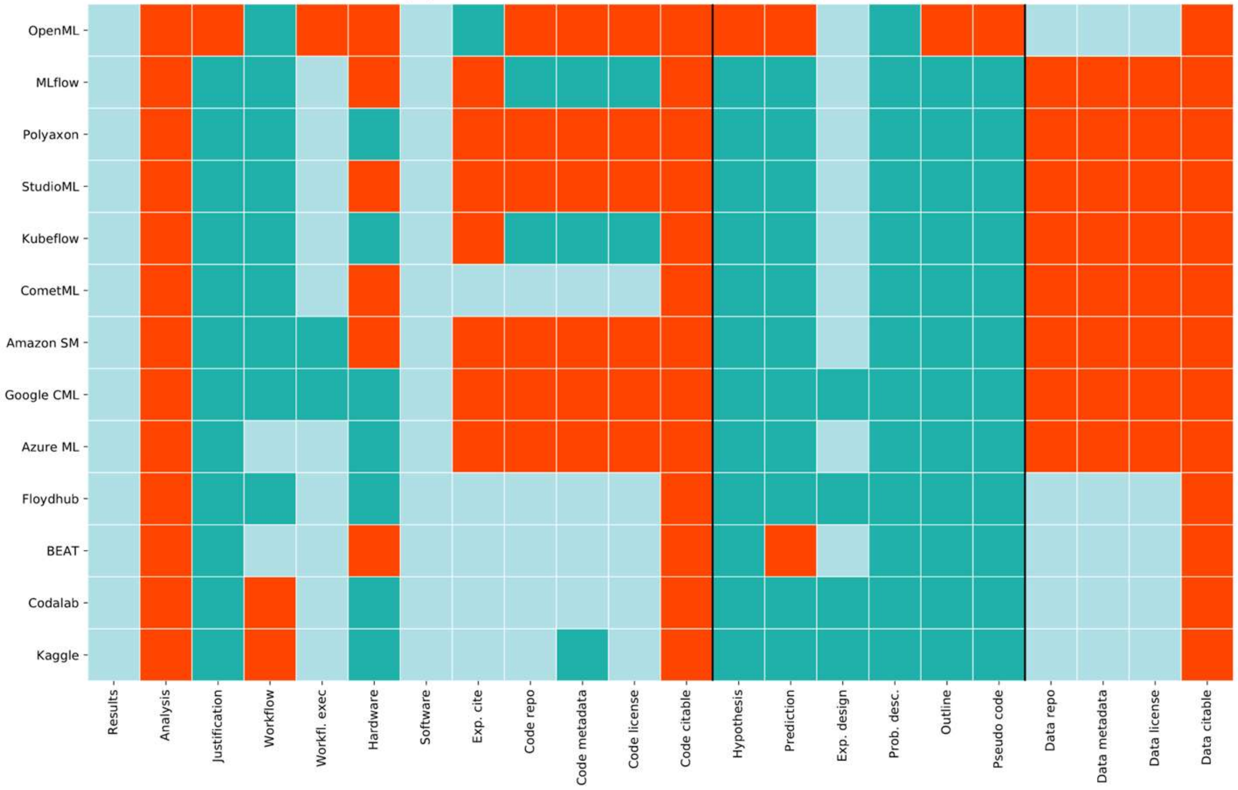
Albertoni et al.

“Do machine learning platforms provide out-of-the-box reproducibility?” [GSI22]

Related Work

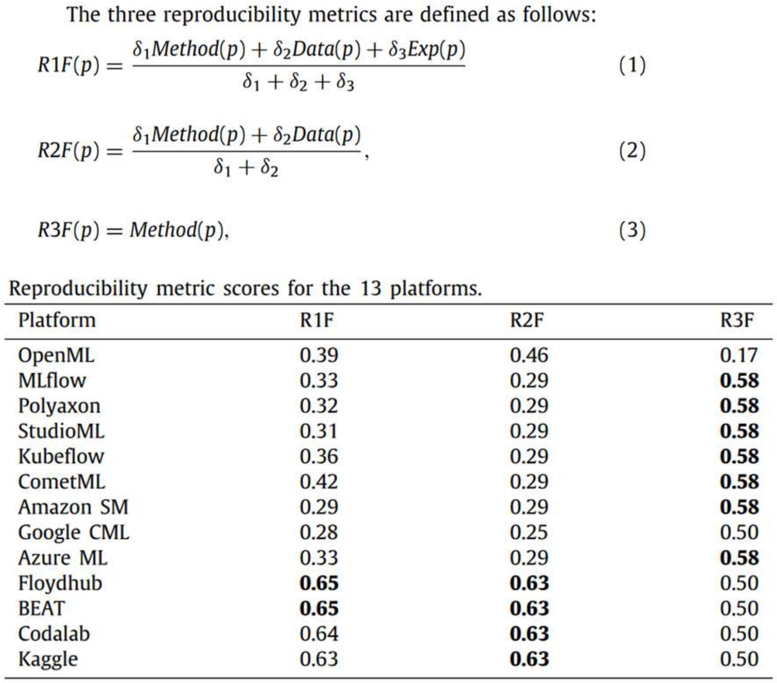
Gundersen et al.

Albertoni et al.



Method Experiment Data

“Do machine learning platforms provide out-of-the-box reproducibility?” [GSI22]



The three reproducibility metrics are defined as follows:

$$R1F(p) = \frac{\delta_1 Method(p) + \delta_2 Data(p) + \delta_3 Exp(p)}{\delta_1 + \delta_2 + \delta_3} \tag{1}$$

$$R2F(p) = \frac{\delta_1 Method(p) + \delta_2 Data(p)}{\delta_1 + \delta_2}, \tag{2}$$

$$R3F(p) = Method(p), \tag{3}$$

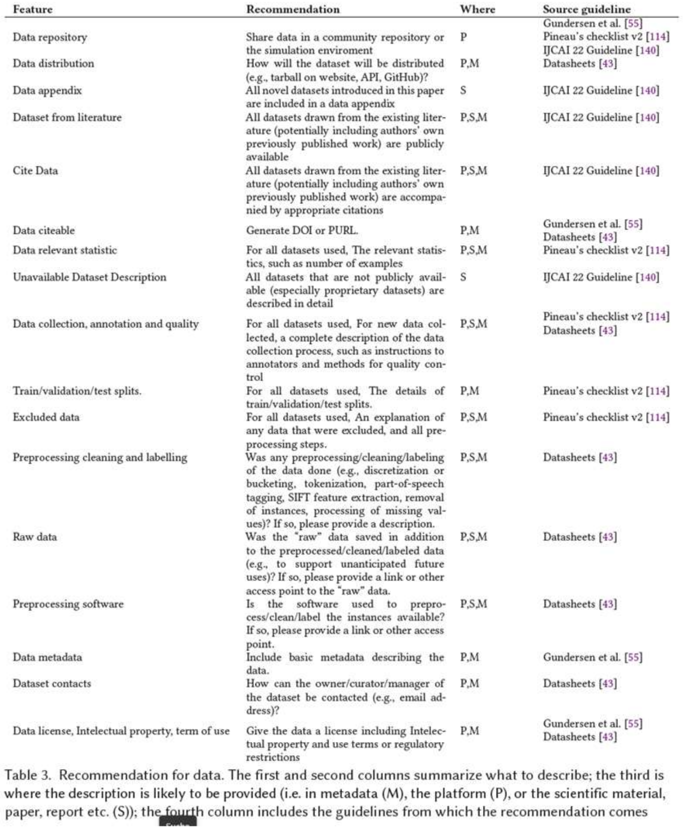
Reproducibility metric scores for the 13 platforms.

Platform	R1F	R2F	R3F
OpenML	0.39	0.46	0.17
MLflow	0.33	0.29	**0.58**
Polyaxon	0.32	0.29	**0.58**
StudioML	0.31	0.29	**0.58**
Kubeflow	0.36	0.29	**0.58**
CometML	0.42	0.29	**0.58**
Amazon SM	0.29	0.29	**0.58**
Google CML	0.28	0.25	0.50
Azure ML	0.33	0.29	**0.58**
Floydhub	**0.65**	**0.63**	0.50
BEAT	**0.65**	**0.63**	0.50
Codalab	0.64	**0.63**	0.50
Kaggle	0.63	**0.63**	0.50

Related Work

Gundersen et al.

Albertoni et al.



Feature	Recommendation	Where	Source guideline
Data repository	Share data in a community repository or the simulation enviroment	P	Gundersen et al. [55] Pineau's checklist v2 [114] IJCAI 22 Guideline [140]
Data distribution	How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?	P,M	Datasheets [43]
Data appendix	All novel datasets introduced in this paper are included in a data appendix	S	IJCAI 22 Guideline [140]
Dataset from literature	All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available	P,S,M	IJCAI 22 Guideline [140]
Cite Data	All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations	P,S,M	IJCAI 22 Guideline [140]
Data citeable	Generate DOI or PURL.	P,M	Gundersen et al. [55] Datasheets [43]
Data relevant statistic	For all datasets used, The relevant statistics, such as number of examples	P,S,M	Pineau's checklist v2 [114]
Unavailable Dataset Description	All datasets that are not publicly available (especially proprietary datasets) are described in detail	S	IJCAI 22 Guideline [140]
Data collection, annotation and quality	For all datasets used, For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control	P,S,M	Pineau's checklist v2 [114] Datasheets [43]
Train/validation/test splits.	For all datasets used, The details of train/validation/test splits.	P,M	Pineau's checklist v2 [114]
Excluded data	For all datasets used, An explanation of any data that were excluded, and all preprocessing steps.	P,S,M	Pineau's checklist v2 [114]
Preprocessing cleaning and labelling	Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description.	P,S,M	Datasheets [43]
Raw data	Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.	P,S,M	Datasheets [43]
Preprocessing software	Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.	P,S,M	Datasheets [43]
Data metadata	Include basic metadata describing the data.	P,M	Gundersen et al. [55]
Dataset contacts	How can the owner/curator/manager of the dataset be contacted (e.g., email address)?	P,M	Datasheets [43]
Data license, Intelectual property, term of use	Give the data a license including Intelectual property and use terms or regulatory restrictions	P,M	Gundersen et al. [55] Datasheets [43]

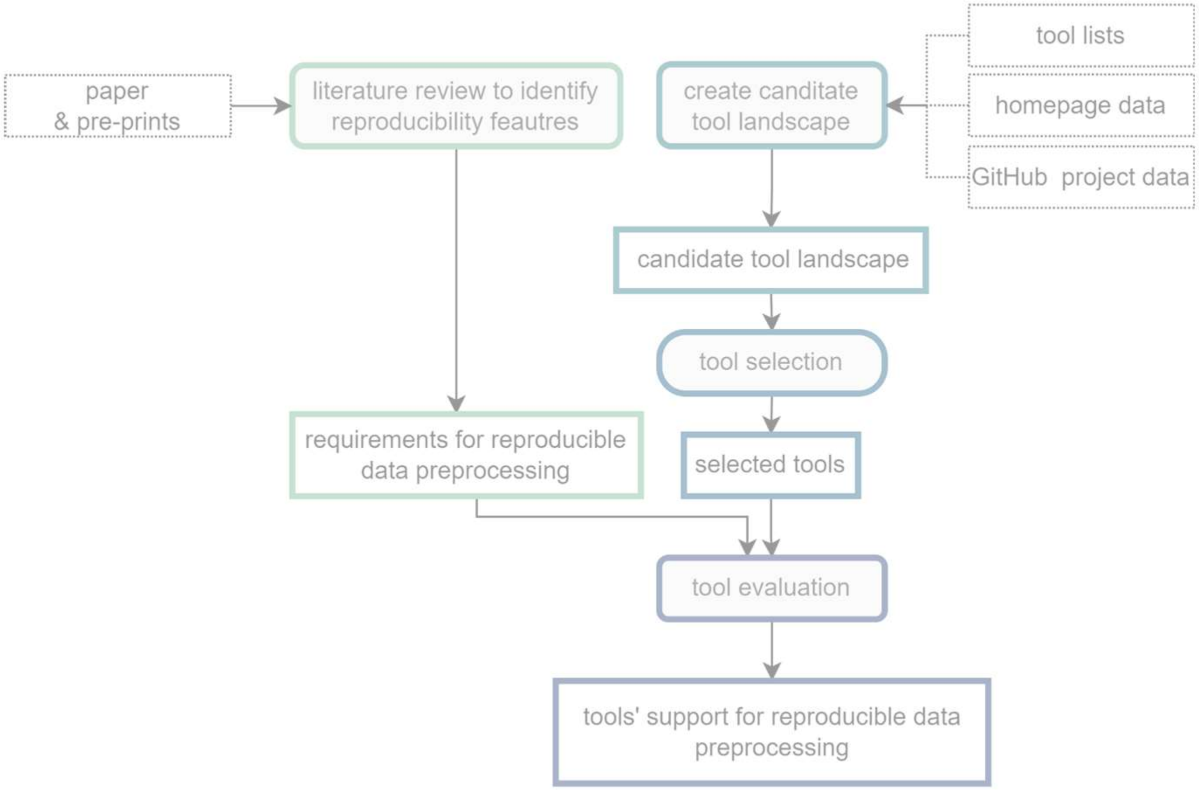
Table 3. Recommendation for data. The first and second columns summarize what to describe; the third is where the description is likely to be provided (i.e. in metadata (M), the platform (P), or the scientific material, paper, report etc. (S)); the fourth column includes the guidelines from which the recommendation comes

"Reproducibility of Machine Learning:

Terminology, Recommendations and Open Issues" [Alb+23].

Methods and Design

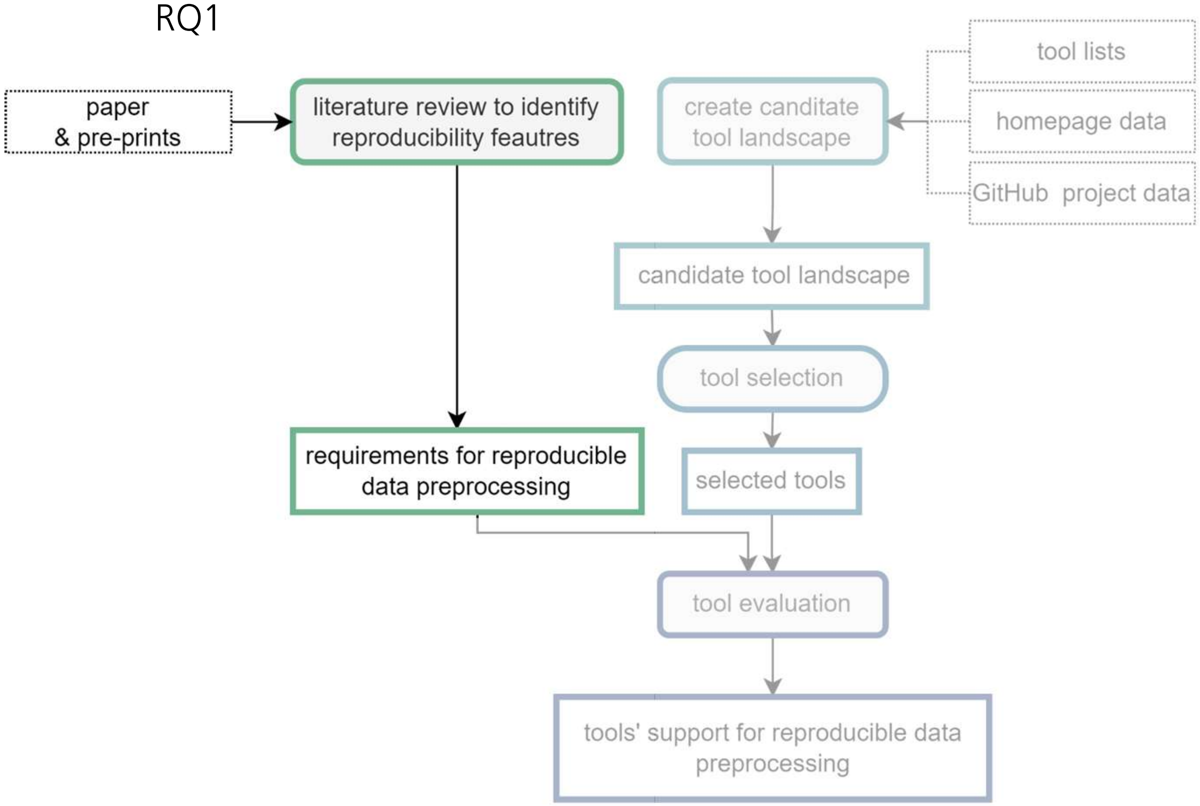
Overview   Requirements for Reproducibility   Candidate Tool Landscape   Tool Selection   Tool Evaluation


paper
& pre-prints
literature review to identify reproducibility feautres
create canditate tool landscape
tool lists
homepage data
GitHub project data
candidate tool landscape
tool selection
requirements for reproducible data preprocessing
selected tools
tool evaluation
tools' support for reproducible data preprocessing

Methods and Design

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation

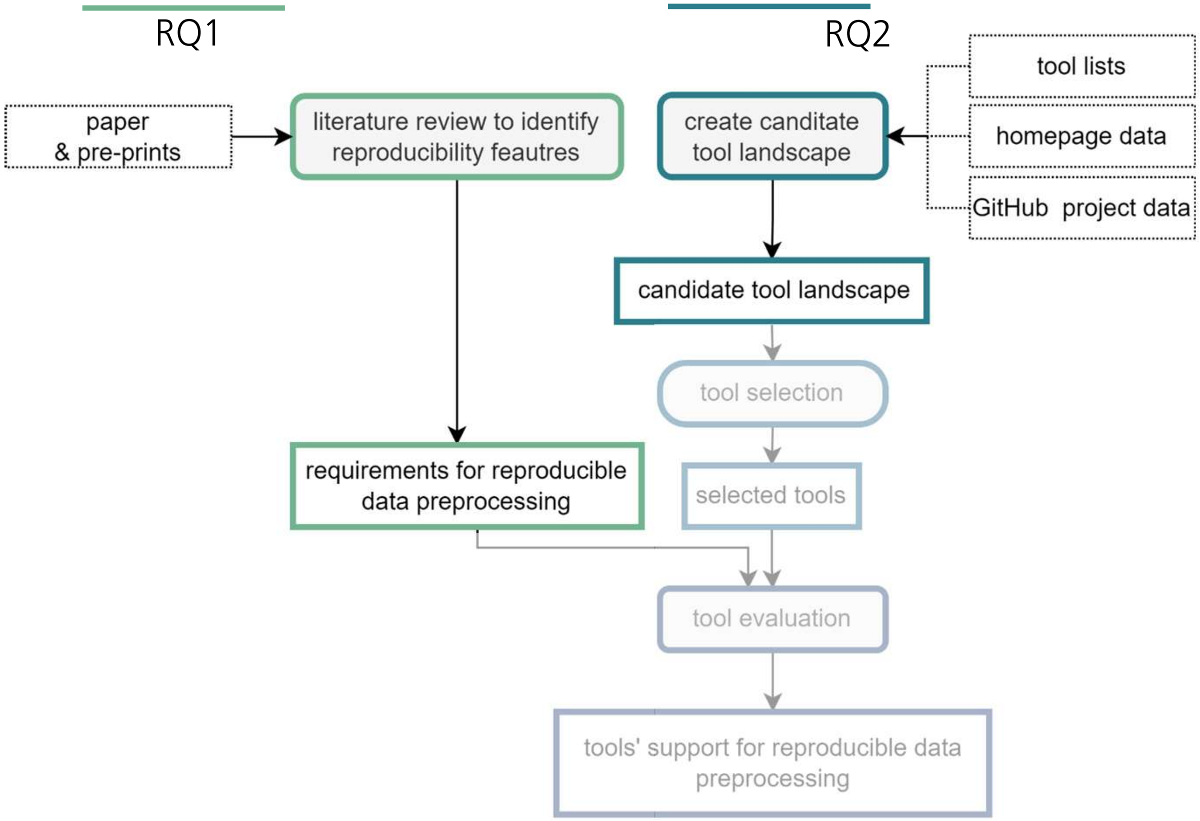
RQ1



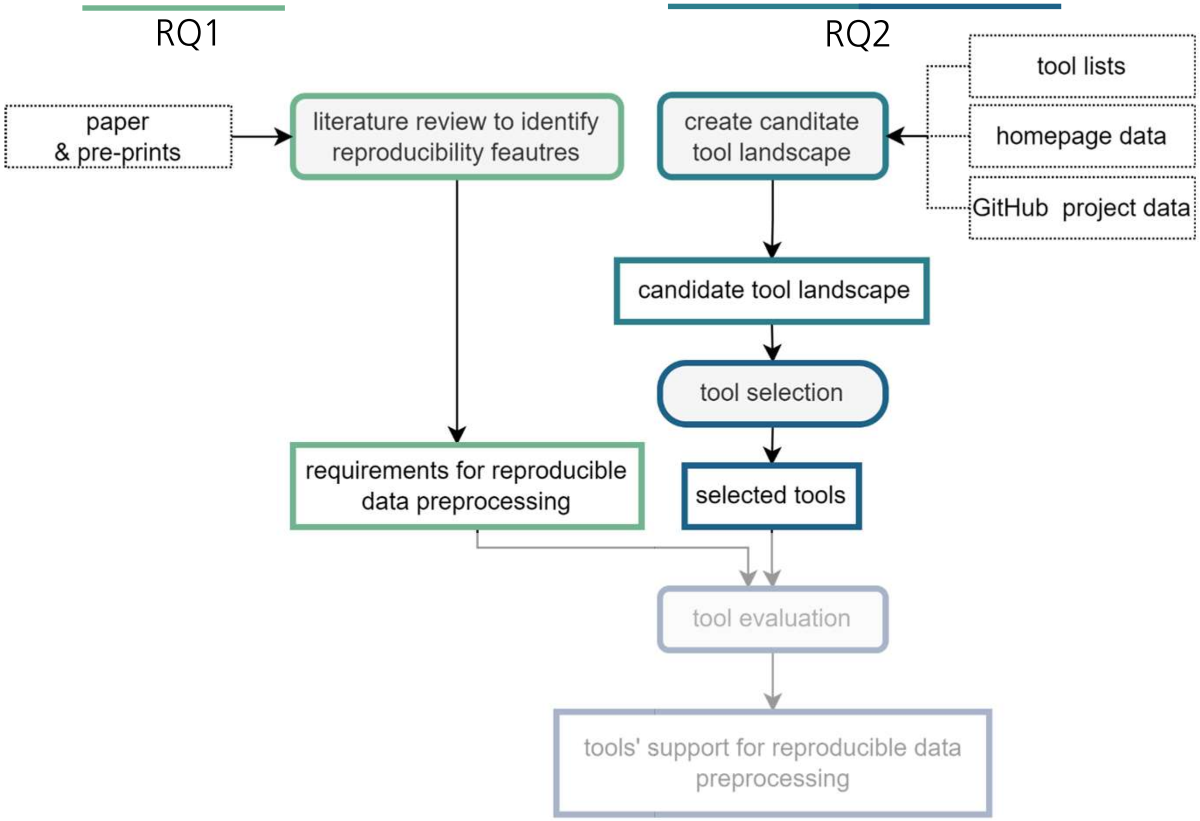
paper
& pre-prints
literature review to identify reproducibility feautres
requirements for reproducible data preprocessing
tool lists
homepage data
GitHub project data
create canditate tool landscape
candidate tool landscape
tool selection
selected tools
tool evaluation
tools' support for reproducible data preprocessing

Methods and Design

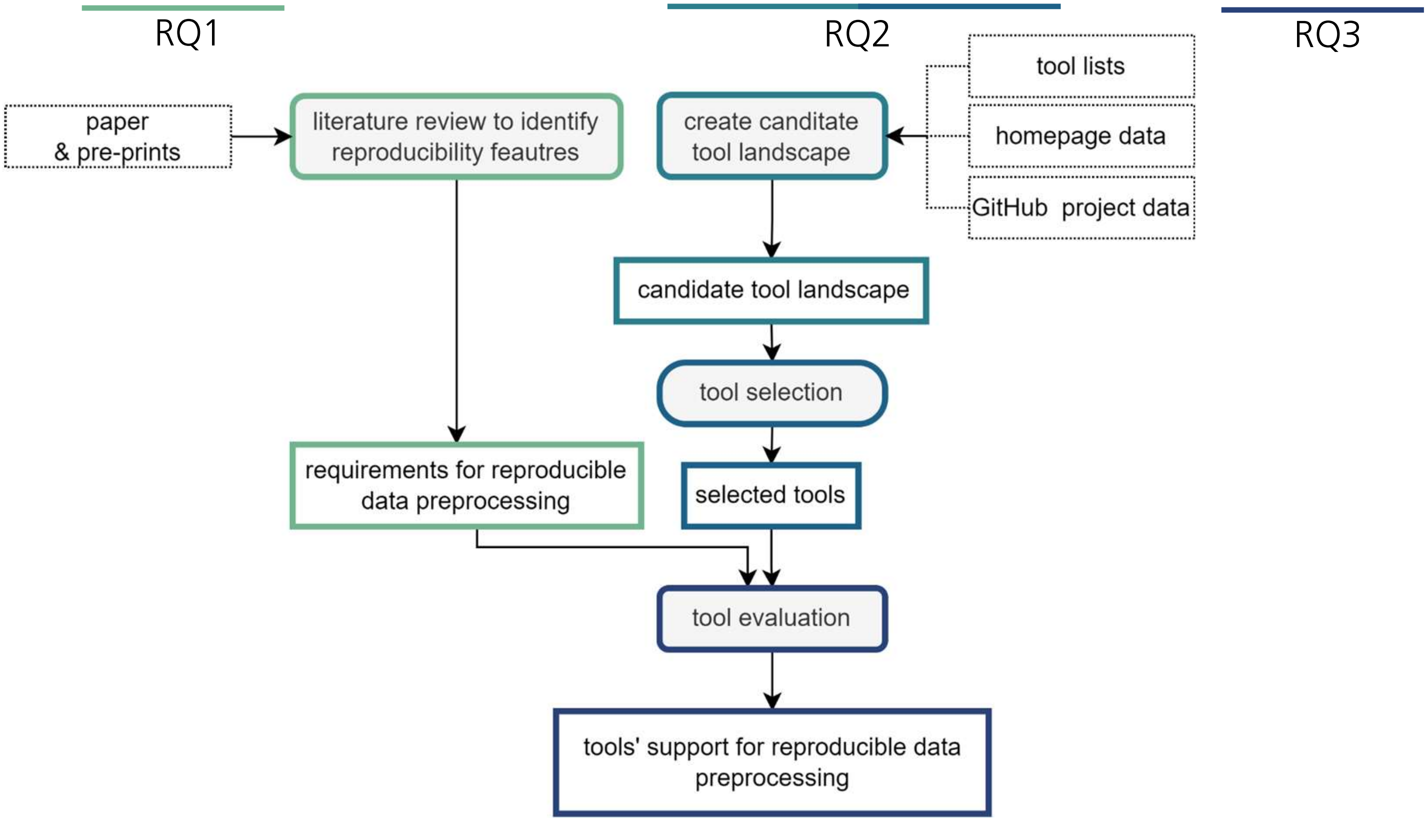
Overview Requirements for Reproducibility Candidate Tool Landscape Tool Selection Tool Evaluation


RQ1
RQ2
tool lists
paper & pre-prints
literature review to identify reproducibility feautres
create canditate tool landscape
homepage data
GitHub project data
candidate tool landscape
tool selection
requirements for reproducible data preprocessing
selected tools
tool evaluation
tools' support for reproducible data preprocessing

Methods and Design


RQ1
RQ2
paper
& pre-prints
literature review to identify reproducibility feautres
requirements for reproducible data preprocessing
tool lists
homepage data
GitHub project data
create canditate tool landscape
candidate tool landscape
tool selection
selected tools
tool evaluation
tools' support for reproducible data preprocessing

Methods and Design



RQ1

RQ2

RQ3

paper & pre-prints

literature review to identify reproducibility feautres

requirements for reproducible data preprocessing

tool lists

homepage data

GitHub project data

create canditate tool landscape

candidate tool landscape

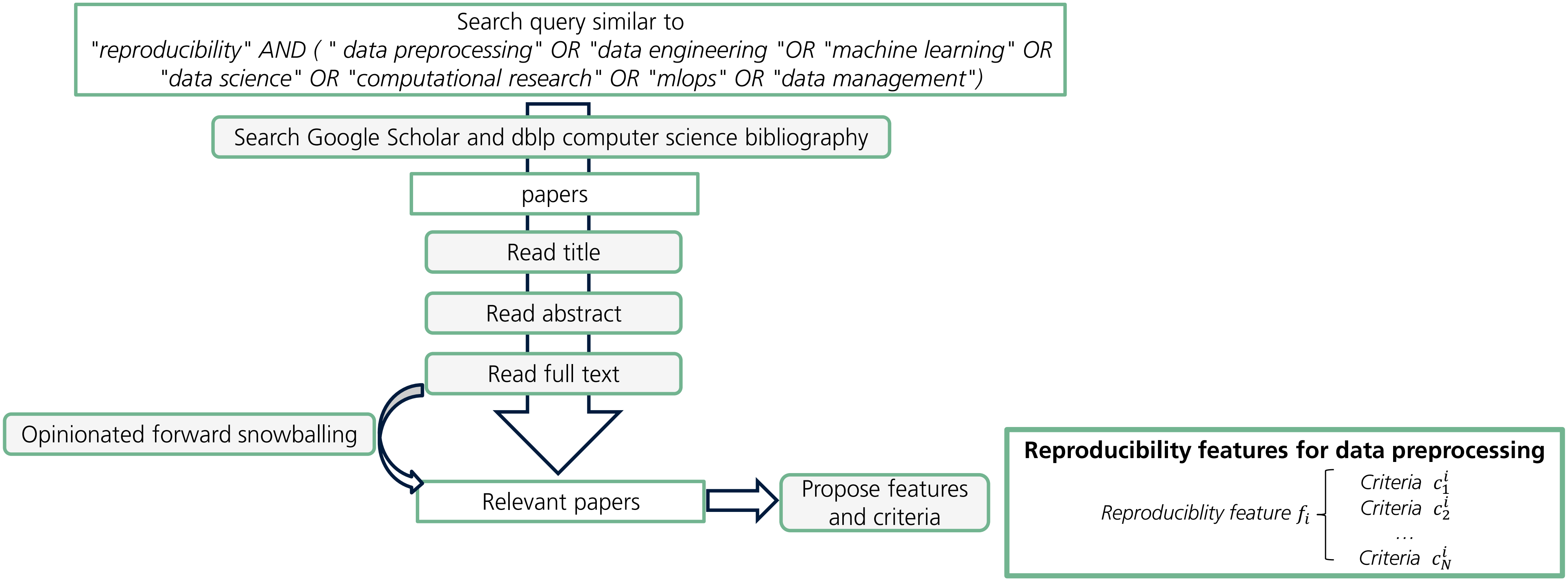
tool selection

selected tools

tool evaluation

tools' support for reproducible data preprocessing

Methods and Design



Search query similar to
"*reproducibility*" *AND* (" *data preprocessing*" *OR* "*data engineering* "*OR* "*machine learning*" *OR* "*data science*" *OR* "*computational research*" *OR* "*mlops*" *OR* "*data management*")

Search Google Scholar and dblp computer science bibliography

papers

Read title

Read abstract

Read full text

Opinionated forward snowballing

Relevant papers

Propose features and criteria

Reproducibility features for data preprocessing

$$\textit{Reproduciblity feature } f_i \left\{ \begin{array}{l} \textit{Criteria } c_1^i \\ \textit{Criteria } c_2^i \\ \quad \dots \\ \textit{Criteria } c_N^i \end{array} \right.$$

Methods and Design

Overview | Requirements for Reproducibility | Candidate Tool Landscape ▲ | Tool Selection | Tool Evaluation

Approach

Use community-maintained lists of tools in the AI and data domain to create a candidate tool landscape, and integrate GitHub and homepage data.

Methods and Design

Raw Data Sources

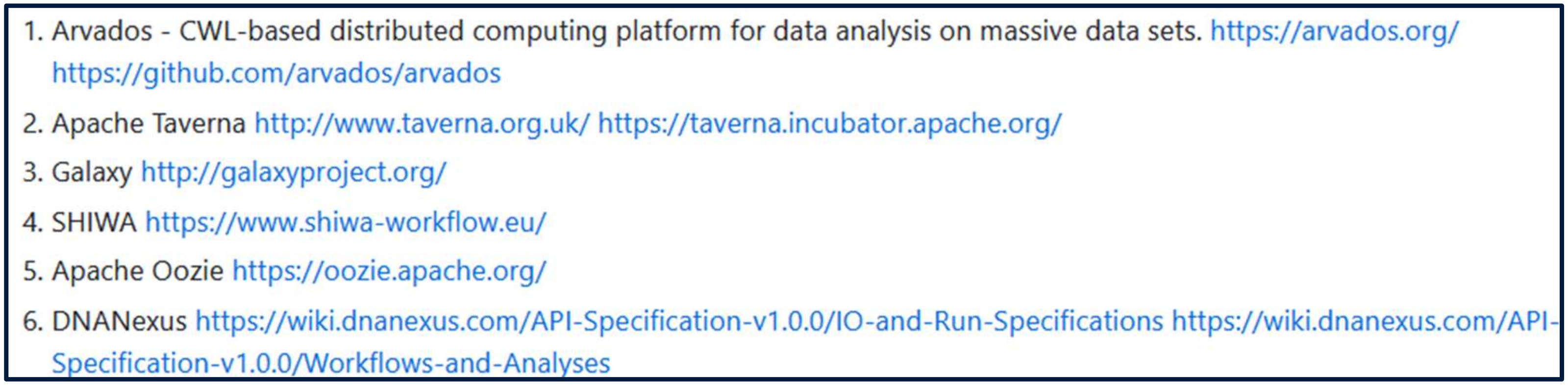
Existing Workflow Systems - https://s.apache.org/existing-workflow-systems

- incomplete list of computational analysis workflow systems
- Information per tool:
  - Tool name
  - Description (optional)
  - One or more Uniform Resource Locators (URLs) to the tool homepage, repository, or publication
- File format:
  - reStructuredText

Methods and Design

Raw Data Sources

Existing Workflow Systems - https://s.apache.org/existing-workflow-systems



1. Arvados - CWL-based distributed computing platform for data analysis on massive data sets. https://arvados.org/ https://github.com/arvados/arvados
2. Apache Taverna http://www.taverna.org.uk/ https://taverna.incubator.apache.org/
3. Galaxy http://galaxyproject.org/
4. SHIWA https://www.shiwa-workflow.eu/
5. Apache Oozie https://oozie.apache.org/
6. DNANexus https://wiki.dnanexus.com/API-Specification-v1.0.0/IO-and-Run-Specifications https://wiki.dnanexus.com/API-Specification-v1.0.0/Workflows-and-Analyses

Methods and Design

Overview Requirements for Reproducibility Candidate Tool Landscape ▲ Tool Selection Tool Evaluation

Raw Data Sources

Linux Foundation AI and Data Landscape- https://landscape.lfai.foundation/

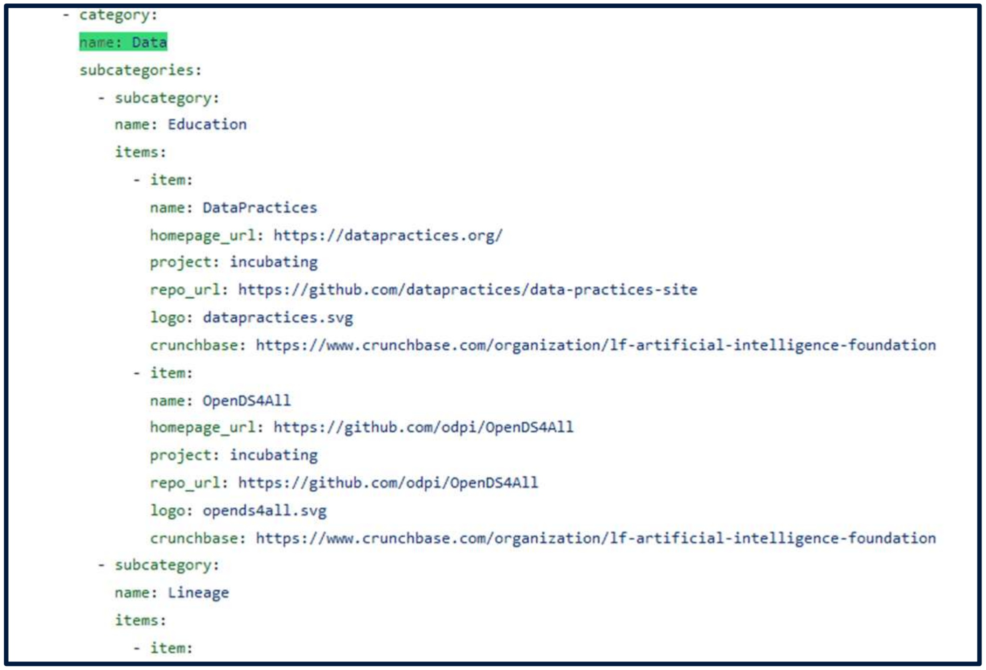
- Interactive tool overview in the AI and data domain.
- Landscape is dynamically generated based on a YAML file in the corresponding GitHub repository
- Relevant information per tool:
    - Tool name
    - Category and Subcategory
    - Homepage URL
    - Repository URL
- File format:
    - YAML

Methods and Design

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation

Raw Data Sources

Linux Foundation AI and Data Landscape- https://landscape.lfai.foundation/



```
- category:
  name: Data
  subcategories:
    - subcategory:
      name: Education
      items:
        - item:
          name: DataPractices
          homepage_url: https://datapractices.org/
          project: incubating
          repo_url: https://github.com/datapractices/data-practices-site
          logo: datapractices.svg
          crunchbase: https://www.crunchbase.com/organization/lf-artificial-intelligence-foundation
        - item:
          name: OpenDS4All
          homepage_url: https://github.com/odpi/OpenDS4All
          project: incubating
          repo_url: https://github.com/odpi/OpenDS4All
          logo: opends4all.svg
          crunchbase: https://www.crunchbase.com/organization/lf-artificial-intelligence-foundation
    - subcategory:
      name: Lineage
      items:
        - item:
```

Methods and Design

Raw Data Sources

Awesome Pipeline- https://github.com/pditommaso/awesome-pipeline

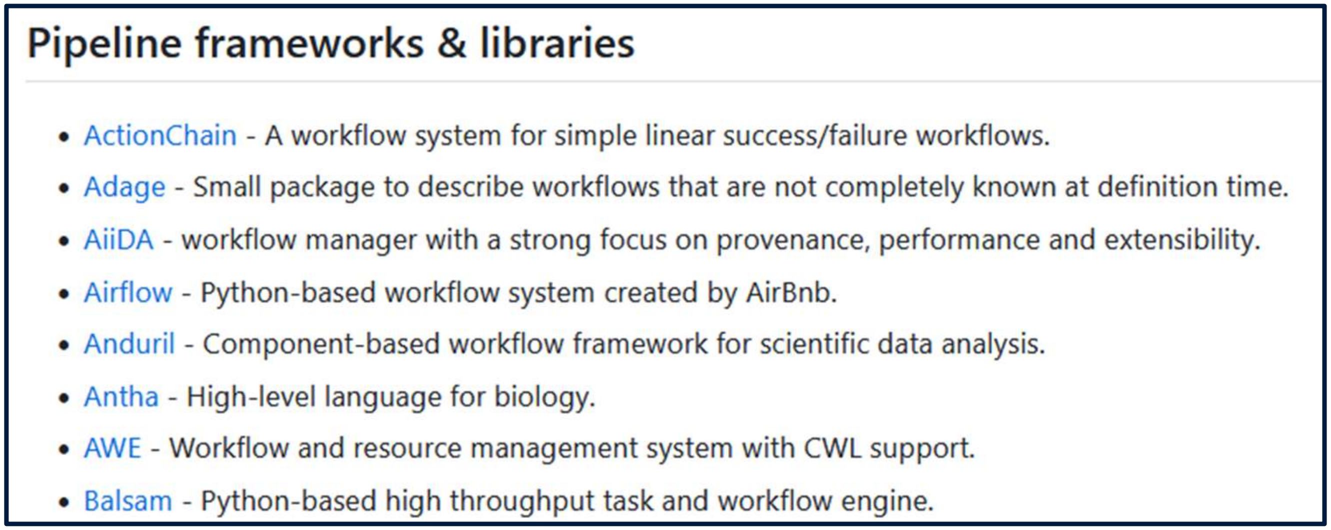
- Community-curated list focusing on pipeline toolkits
- Information per tool:
  - Tool name
  - Short description
  - Grouped via sections and subsections
  - URL – either repository or homepage
- File format:
  - Markdown

Methods and Design

Overview | Requirements for Reproducibility | Candidate Tool Landscape ▲ | Tool Selection | Tool Evaluation

Raw Data Sources

Awesome Pipeline- https://github.com/pditommaso/awesome-pipeline



Pipeline frameworks & libraries

- ActionChain - A workflow system for simple linear success/failure workflows.
- Adage - Small package to describe workflows that are not completely known at definition time.
- AiiDA - workflow manager with a strong focus on provenance, performance and extensibility.
- Airflow - Python-based workflow system created by AirBnb.
- Anduril - Component-based workflow framework for scientific data analysis.
- Antha - High-level language for biology.
- AWE - Workflow and resource management system with CWL support.
- Balsam - Python-based high throughput task and workflow engine.

Methods and Design

Raw Data Sources

Awesome Data Engineering- https://github.com/igorbarinov/awesome-data-engineering

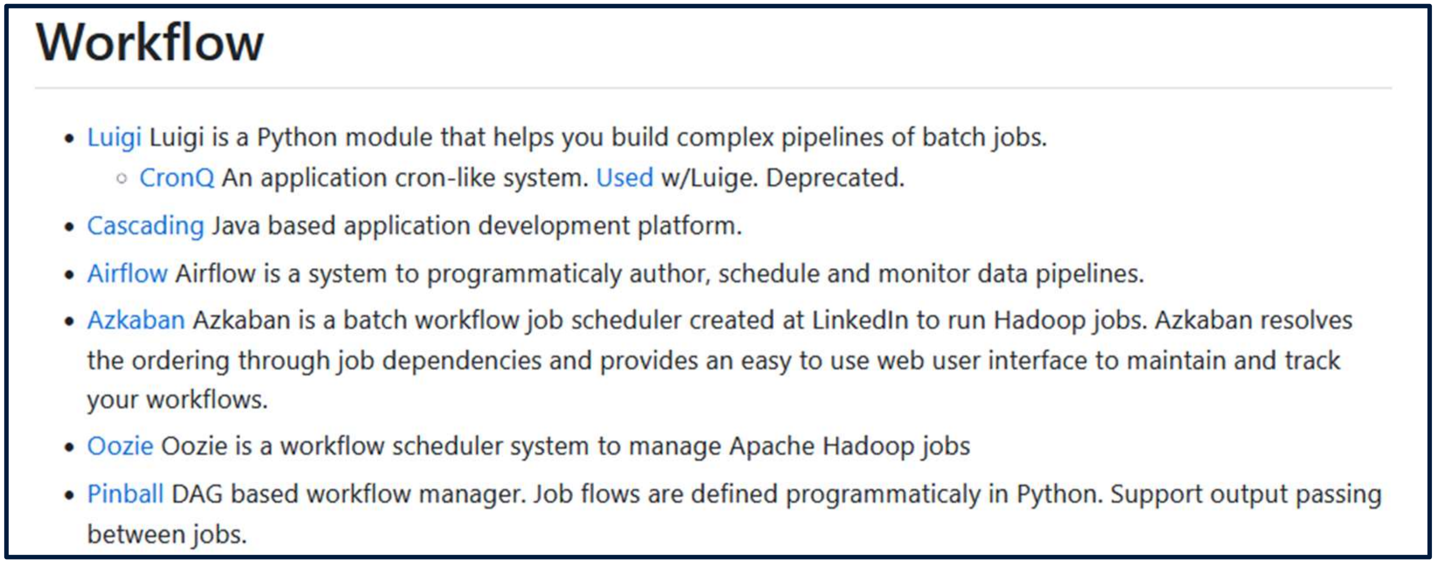
- Community-curated list focusing on data engineering tools
- Information per tool:
  - Tool name
  - Short description
  - Grouped via sections and subsections
  - URL – either repository or homepage
- File format:
  - Markdown

Methods and Design

Overview | Requirements for Reproducibility | Candidate Tool Landscape ▲ | Tool Selection | Tool Evaluation

Raw Data Sources

Awesome Data Engineering- https://github.com/igorbarinov/awesome-data-engineering



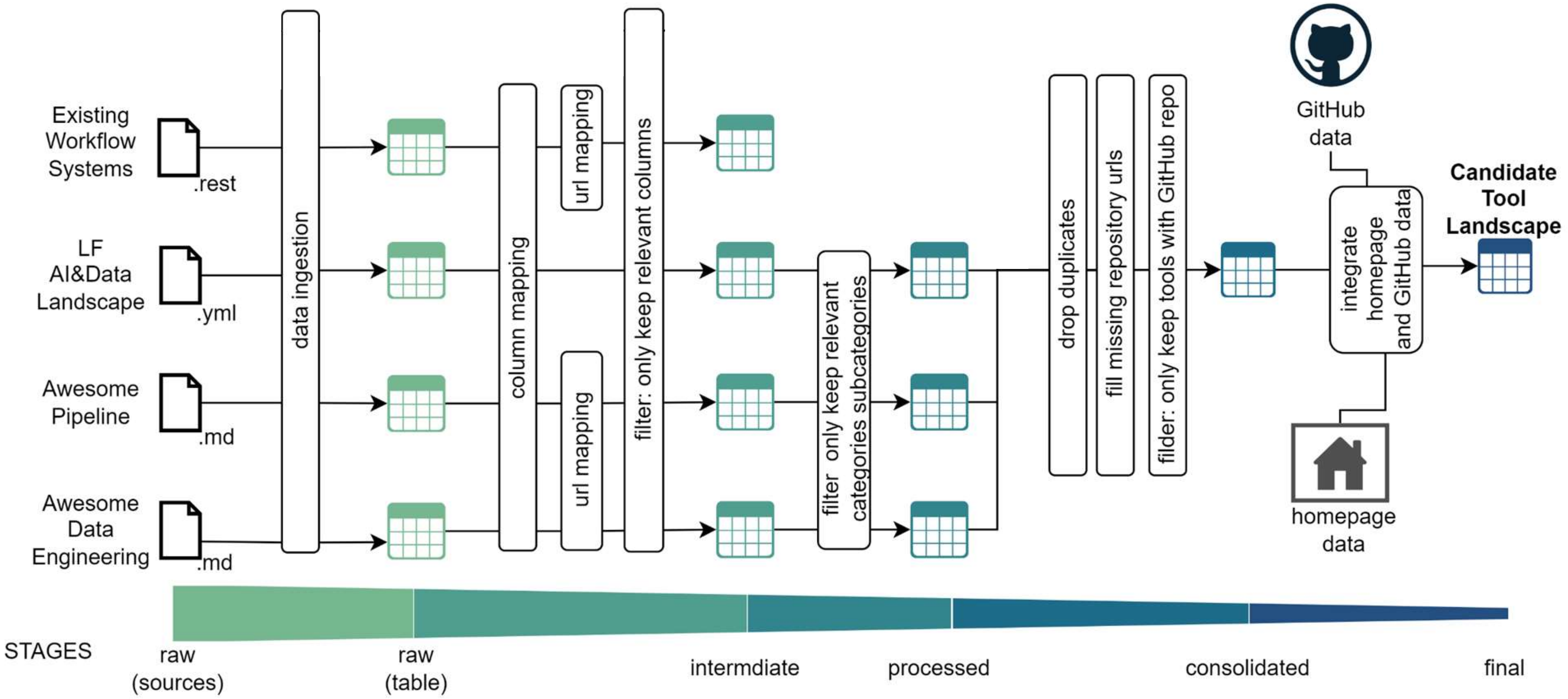
Workflow

- Luigi Luigi is a Python module that helps you build complex pipelines of batch jobs.
  - CronQ An application cron-like system. Used w/Luige. Deprecated.
- Cascading Java based application development platform.
- Airflow Airflow is a system to programmaticaly author, schedule and monitor data pipelines.
- Azkaban Azkaban is a batch workflow job scheduler created at LinkedIn to run Hadoop jobs. Azkaban resolves the ordering through job dependencies and provides an easy to use web user interface to maintain and track your workflows.
- Oozie Oozie is a workflow scheduler system to manage Apache Hadoop jobs
- Pinball DAG based workflow manager. Job flows are defined programmaticaly in Python. Support output passing between jobs.

Methods and Design

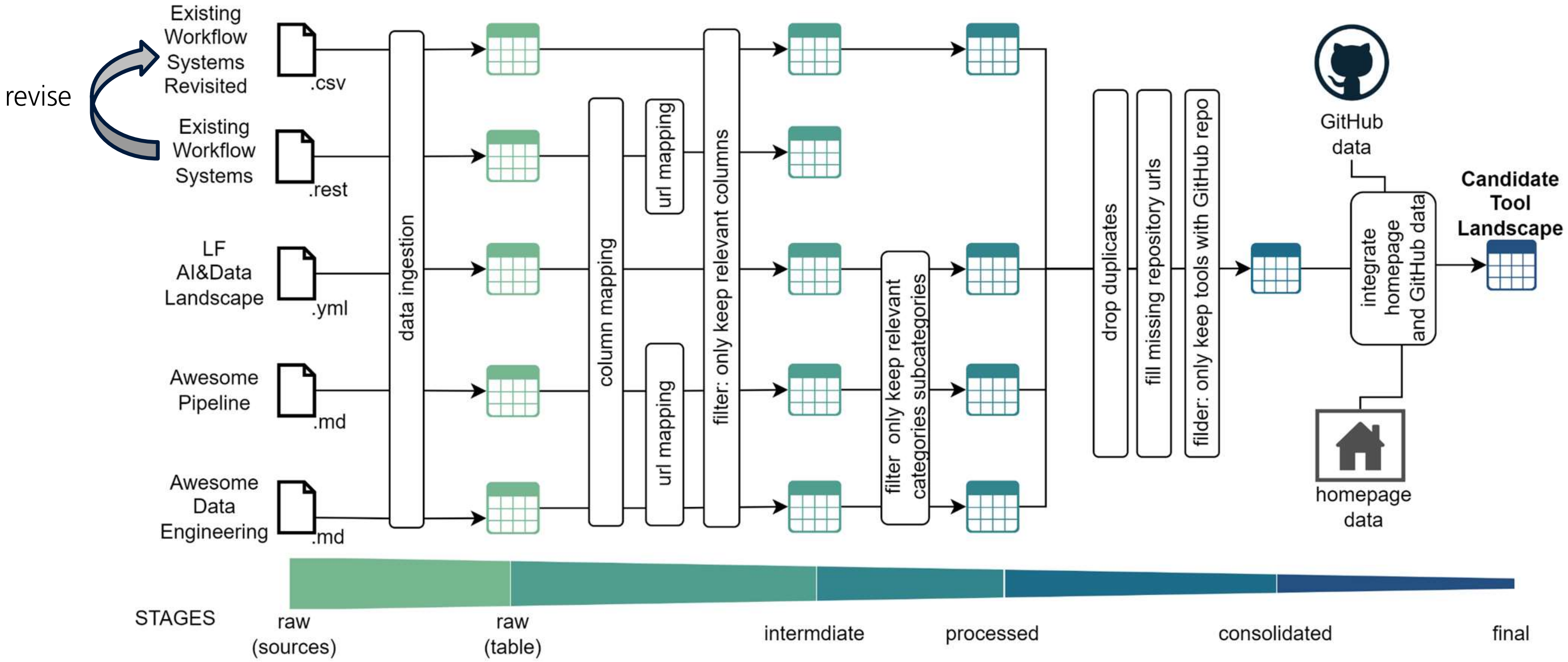
Overview | Requirements for Reproducibility | Candidate Tool Landscape ▲ | Tool Selection | Tool Evaluation

Data preprocessing pipeline to create the candidate tool landscape



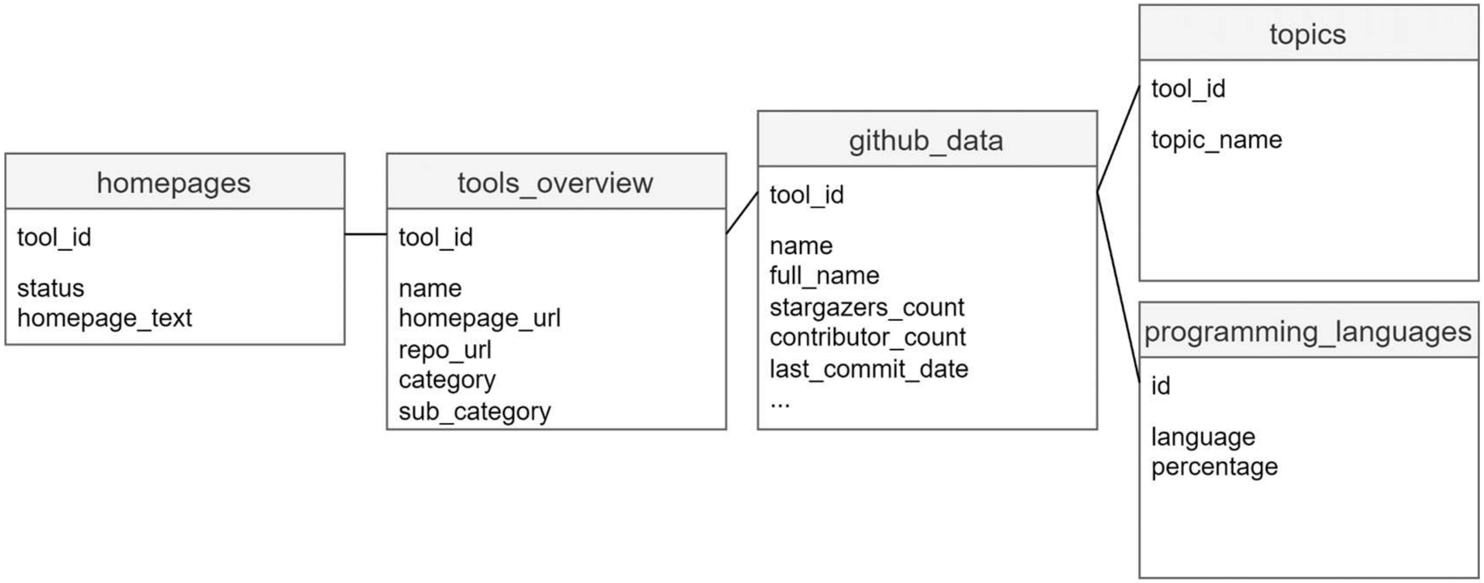
Methods and Design

Data preprocessing pipeline to create the candidate tool landscape



Methods and Design

- Conceptual Data Model



homepages

tool_id

status
homepage_text

tools_overview

tool_id

name
homepage_url
repo_url
category
sub_category

github_data

tool_id

name
full_name
stargazers_count
contributor_count
last_commit_date
...

topics

tool_id

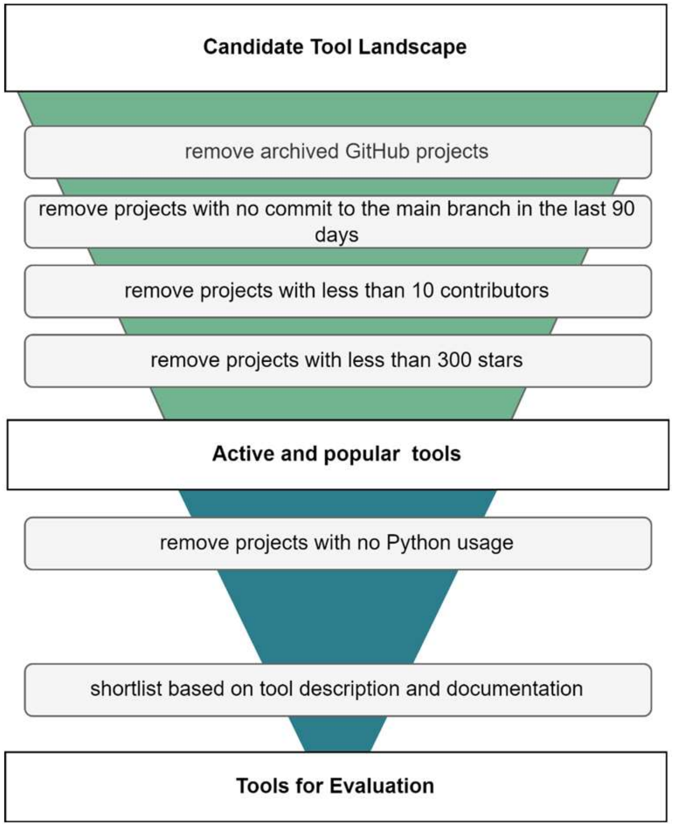
topic_name

programming_languages

id

language
percentage

Methods and Design



Candidate Tool Landscape

remove archived GitHub projects

remove projects with no commit to the main branch in the last 90 days

remove projects with less than 10 contributors

remove projects with less than 300 stars

Active and popular tools

remove projects with no Python usage

shortlist based on tool description and documentation

Tools for Evaluation

Methods and Design

Overview Requirements for Reproducibility Candidate Tool Landscape Tool Selection Tool Evaluation ▲

1. Search the documentation and homepage for each selected tool to identify the support for each criterion of a reproducibility feature by assigning the support level.

Methods and Design

1. Search the documentation and homepage for each selected tool to identify the support for each criterion of a reproducibility feature by assigning the support level.
    1. unsupported The tool does not support the criterion of a reproducibility feature.
    2. standard solution The tool does not support the criterion of a reproducibility feature. However, this gap can be closed by a solution that the community sees as a default. For example, GitHub for public code hosting
    3. enterprise support The tool supports the criterion of a reproducibility feature in the enterprise version, but the functionality is not available in the open-source version.
    4. integration The tool proves an integration with a third-party solution, which supports the criterion of a reproducibility feature.
    5. partially The tool partially supports the criterion of a reproducibility feature.
    6. full The tool partially supports the criterion of a reproducibility feature.

Methods and Design

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation ▲

1. Search the documentation and homepage for each selected tool to identify the support for each criterion of a reproducibility feature by assigning the support level.
2. Quantify the support for a reproducibility feature

Methods and Design

1. Search the documentation and homepage for each selected tool to identify the support for each criterion of a reproducibility feature by assigning the support level.
2. Quantify the support for a reproducibility feature
   1. Assign a numeric value to each support level

1. unsupported	0
2. standard solution	0
3. enterprise support	0
4. integration	1
5. partially	0
6. full	1

Methods and Design

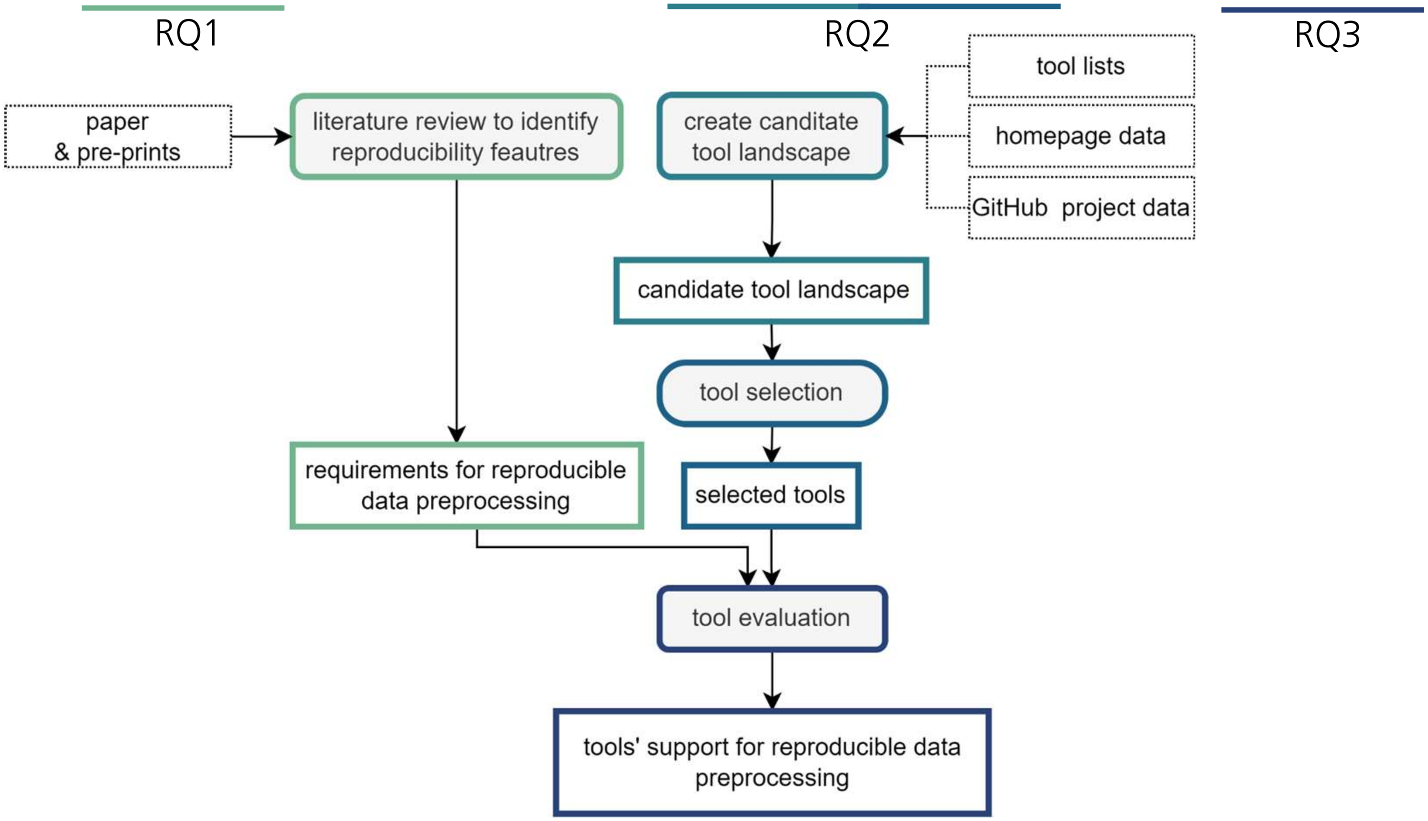
Overview Requirements for Reproducibility Candidate Tool Landscape Tool Selection Tool Evaluation

1. Search the documentation and homepage for each selected tool to identify the support for each criterion of a reproducibility feature by assigning the support level.
2. Quantify the support for a reproducibility feature
   1. Assign a numeric value to each support level
   2. Calculate mean of all criteria values for a feature

Methods and Design

1. Search the documentation and homepage for each selected tool to identify the support for each criterion of a reproducibility feature by assigning the support level.
2. Quantify the support for a reproducibility feature
    1. Assign a numeric value to each support level
    2. Calculate mean of all criteria values for a feature
3. Quantify the overall support for reproducible data preprocessing for each tool
    1. Calculate the mean of all reproducibility feature scores

Results


RQ1
RQ2
RQ3
paper
& pre-prints
literature review to identify reproducibility feautres
requirements for reproducible data preprocessing
tool lists
homepage data
GitHub project data
create canditate tool landscape
candidate tool landscape
tool selection
selected tools
tool evaluation
tools' support for reproducible data preprocessing

Results

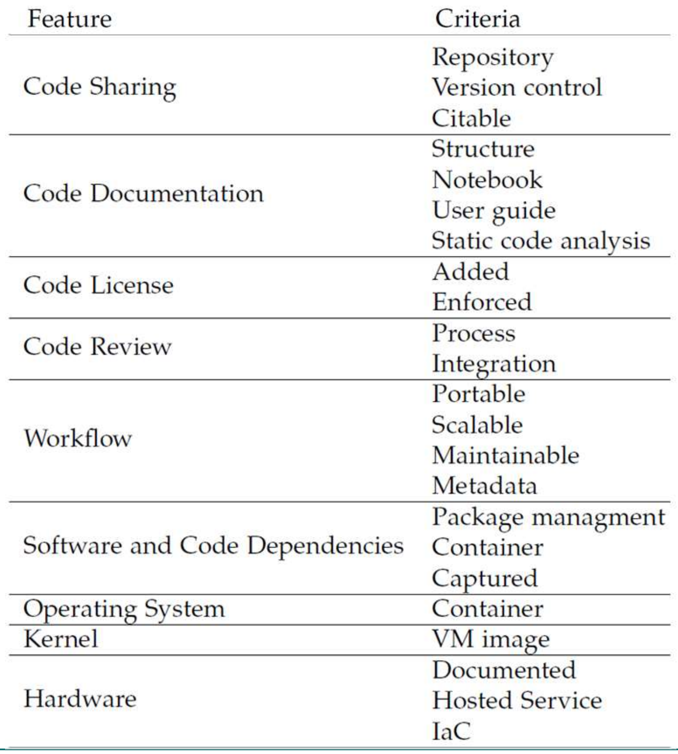
Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation

Proposed reproducibility features for data preprocessing

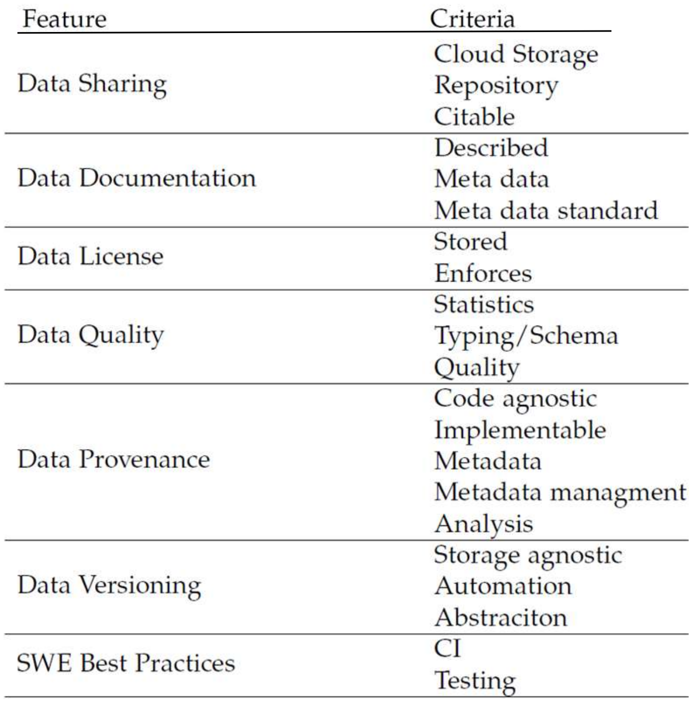
Feature	Nr of criteria	Description
Code Sharing	3	Code is shared in a public code repository, versioned, and is citable.
Code Documentation	4	Code is documented and facilitated by a default structure and notebooks. User guides and static code analysis are supported.
Code License	2	A license is added to the project.
Code Review	2	A Code review process is described or integrated.
Workflow	4	Data preprocessing functions and configurations are abstracted in a workflow representation, such that the workflow is maintainable, portable, scalable, and documented.
Software and Code Dependencies	3	Software and code dependencies are specified in a standardized way using package manager and container.
Operating System	1	Operating System is specified as a part of a container image.
Kernel	1	A virtual machine image can be created.
Hardware	3	The hardware requirements are documented or specified declarative or as infrastructure as code.
SWE best practices	2	Testing and continuous integration are supported.
Data Sharing	3	Data is in a cloud storage, public repository, and is citable.
Data Documentation	3	Data is described in a basic way, annotated, or using a meta data standard.
Data License	2	A license is added to the data.
Data Quality	3	Data quality gates and measures are supported, via statistics, typing, schemas, and advanced data quality assessments.
Data Provenance	5	Data provenance is captured code agnostic, based on workflow implementation. Metadata is captured and stored in metadata management. Analysis of provenance data is supported.
Data Versioning	3	Data is versioned throughout the data lifecycle.

Results

Proposed reproducibility features and criteria for data preprocessing



Feature	Criteria
Code Sharing	Repository Version control Citable
Code Documentation	Structure Notebook User guide Static code analysis
Code License	Added Enforced
Code Review	Process Integration
Workflow	Portable Scalable Maintainable Metadata
Software and Code Dependencies	Package managment Container Captured
Operating System	Container
Kernel	VM image
Hardware	Documented Hosted Service IaC

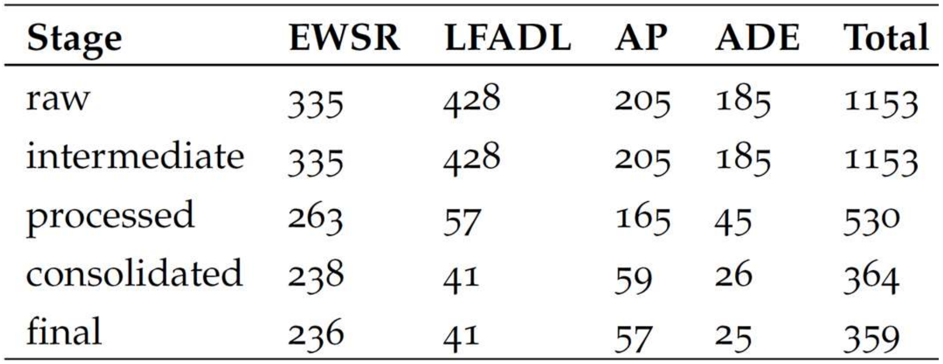


Feature	Criteria
Data Sharing	Cloud Storage Repository Citable
Data Documentation	Described Meta data Meta data standard
Data License	Stored Enforces
Data Quality	Statistics Typing/Schema Quality
Data Provenance	Code agnostic Implementable Metadata Metadata managment Analysis
Data Versioning	Storage agnostic Automation Abstraciton
SWE Best Practices	CI Testing

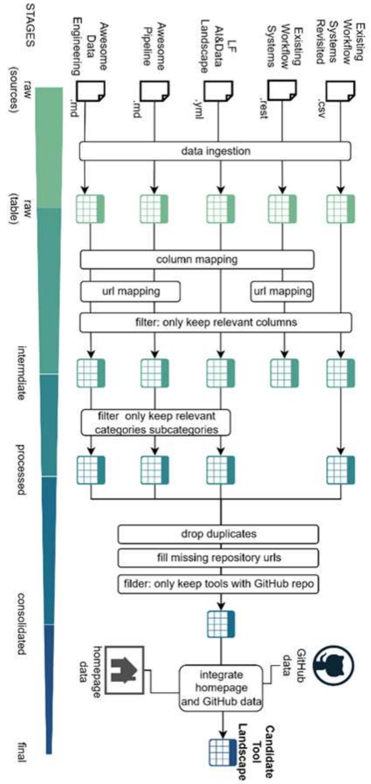
Results

Overview | Requirements for Reproducibility | Candidate Tool Landscape ▲ | Tool Selection | Tool Evaluation

Number of tools for each data source and data preprocessing stage



Stage	EWSR	LFADL	AP	ADE	Total
raw	335	428	205	185	1153
intermediate	335	428	205	185	1153
processed	263	57	165	45	530
consolidated	238	41	59	26	364
final	236	41	57	25	359

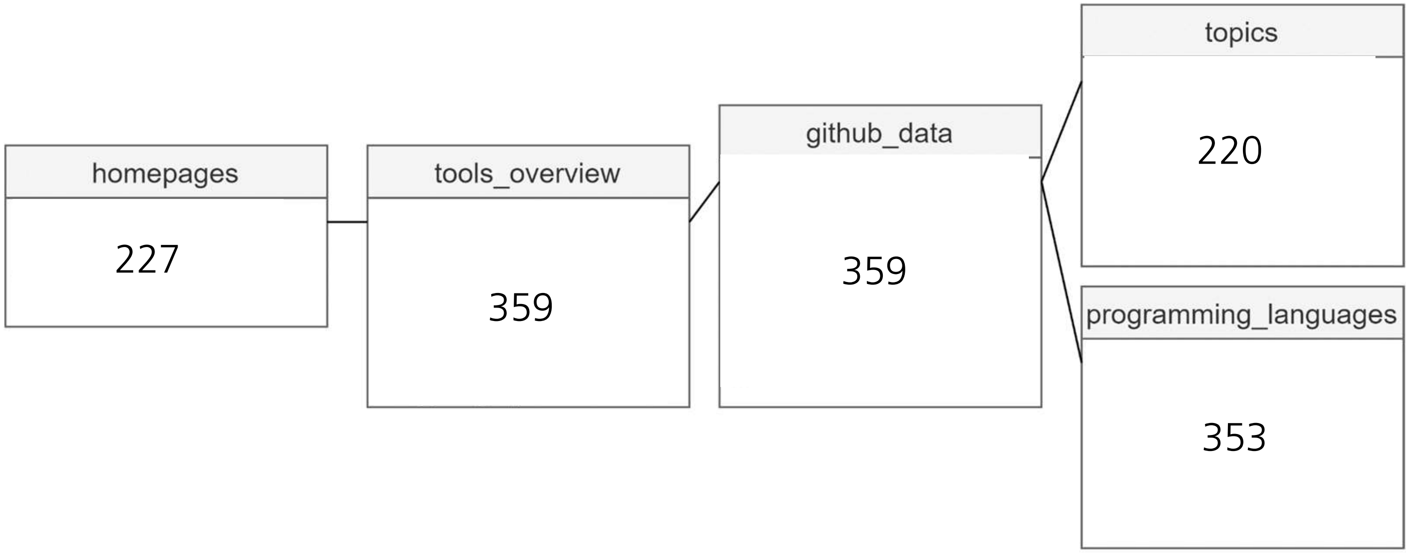


EWSR: Existing Workflow Systems Revisited
LFADL: Linux Foundation AI and Data Landscape
AP: Awesome Pipeline
ADE: Awesome Data Engineering

Results

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation

Number of tools for each table where respective entries are valid and available



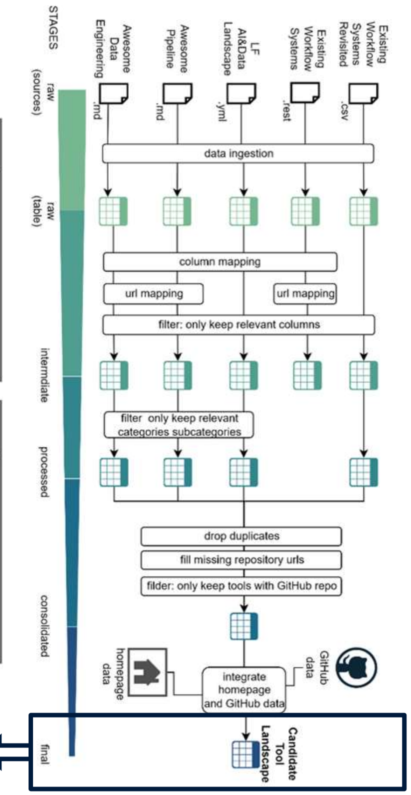
homepages
227

tools_overview
359

github_data
359

topics
220

programming_languages
353

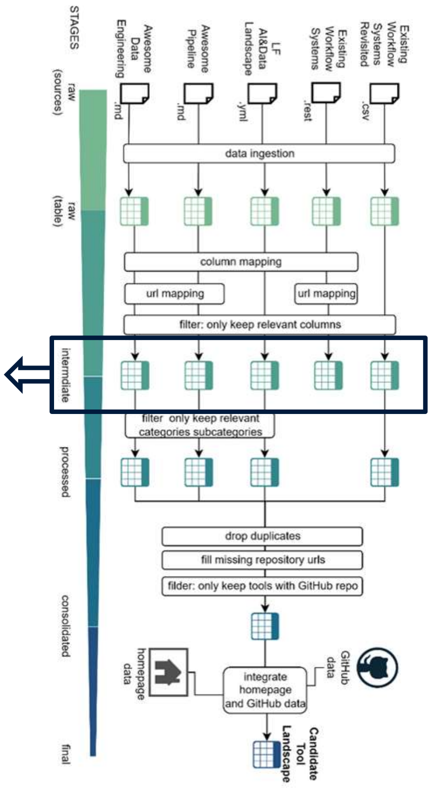


Results

Quality of the raw data sources with respect to provided URLs for the homepage, repository, and publication.

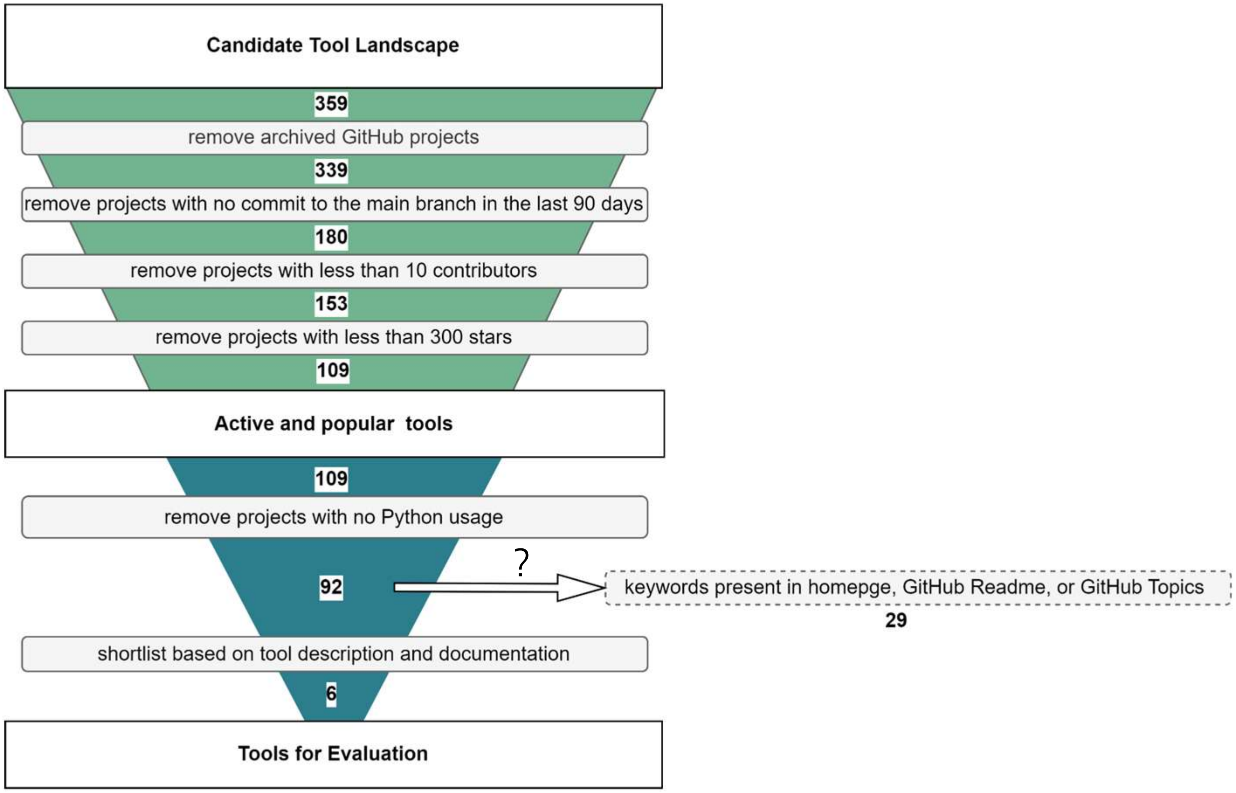
The number of total and valid URLs (200 HTTP responses) is given for the URL columns in the format: nr of URLS (nr of valid URLs).

nr. of	EWSR	EWS	LFADL	AP	ADE
tools	335	335	428	205	185
homepage URLs	252 (229)	209 (186)	428 (406)	89 (77)	106 (104)
publication URLs	199 (191)	75 (70)	0 (0)	0 (0)	0 (0)
repository URLs	263 (259)	158 (155)	340 (340)	116 (115)	79 (77)



EWSR: Existing Workflow Systems Revisited
EWS: Existing Workflow Systems
LFADL: Linux Foundation AI and Data Landscape
AP: Awesome Pipeline
ADE: Awesome Data Engineering

Results



Candidate Tool Landscape

359

remove archived GitHub projects

339

remove projects with no commit to the main branch in the last 90 days

180

remove projects with less than 10 contributors

153

remove projects with less than 300 stars

109

Active and popular tools

109

remove projects with no Python usage

92

?

keywords present in homepge, GitHub Readme, or GitHub Topics

29

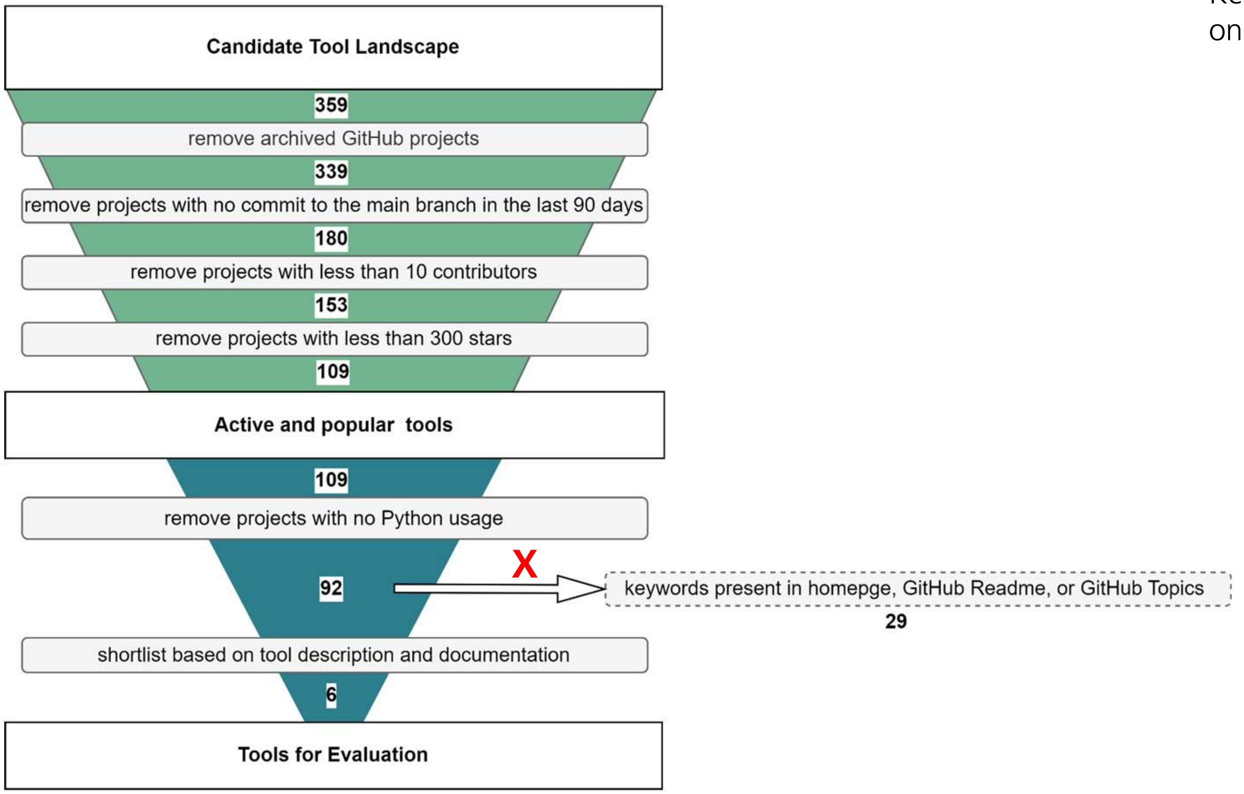
shortlist based on tool description and documentation

6

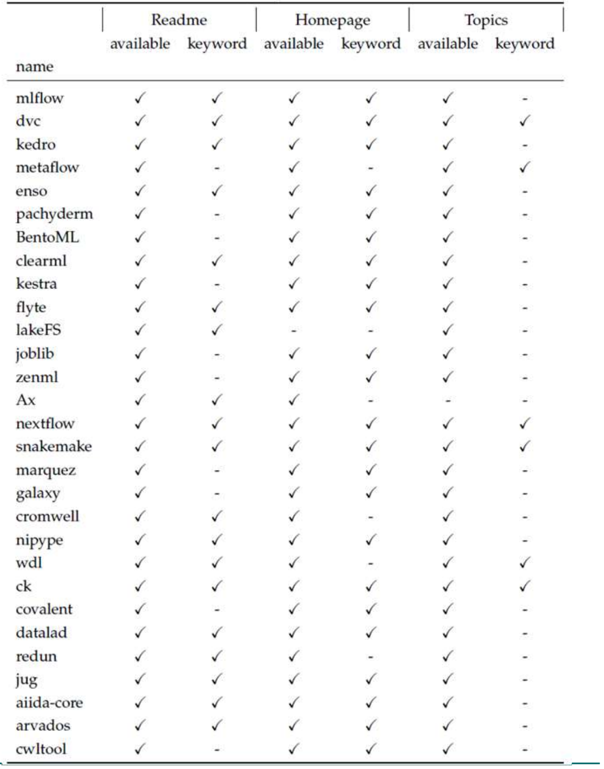
Tools for Evaluation

Results

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation


Candidate Tool Landscape
359
remove archived GitHub projects
339
remove projects with no commit to the main branch in the last 90 days
180
remove projects with less than 10 contributors
153
remove projects with less than 300 stars
109
Active and popular tools
109
remove projects with no Python usage
92
X
keywords present in homepge, GitHub Readme, or GitHub Topics
29
shortlist based on tool description and documentation
6
Tools for Evaluation

Keywords in the “reproducibility” word family occurred on the homepage, GitHub Readme, or GitHub Topics.



	Readme		Homepage		Topics	
	available	keyword	available	keyword	available	keyword
name						
mlflow	✓	✓	✓	✓	✓	-
dvc	✓	✓	✓	✓	✓	✓
kedro	✓	✓	✓	✓	✓	-
metaflow	✓	-	✓	-	✓	✓
enso	✓	✓	✓	✓	✓	-
pachyderm	✓	-	✓	✓	✓	-
BentoML	✓	-	✓	✓	✓	-
clearml	✓	✓	✓	✓	✓	-
kestra	✓	-	✓	✓	✓	-
flyte	✓	✓	✓	✓	✓	-
lakeFS	✓	✓	-	-	✓	-
joblib	✓	-	✓	✓	✓	-
zenml	✓	-	✓	✓	✓	-
Ax	✓	✓	✓	-	-	-
nextflow	✓	✓	✓	✓	✓	✓
snakemake	✓	✓	✓	✓	✓	✓
marquez	✓	-	✓	✓	✓	-
galaxy	✓	-	✓	✓	✓	-
cromwell	✓	✓	✓	-	✓	-
nipype	✓	✓	✓	✓	✓	-
wdl	✓	✓	✓	-	✓	✓
ck	✓	✓	✓	✓	✓	✓
covalent	✓	-	✓	✓	✓	-
datalad	✓	✓	✓	✓	✓	-
redun	✓	✓	✓	-	✓	-
jug	✓	✓	✓	✓	✓	-
aiida-core	✓	✓	✓	✓	✓	-
arvados	✓	✓	✓	✓	✓	-
cwltool	✓	-	✓	✓	✓	-

Results

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection ▲ | Tool Evaluation

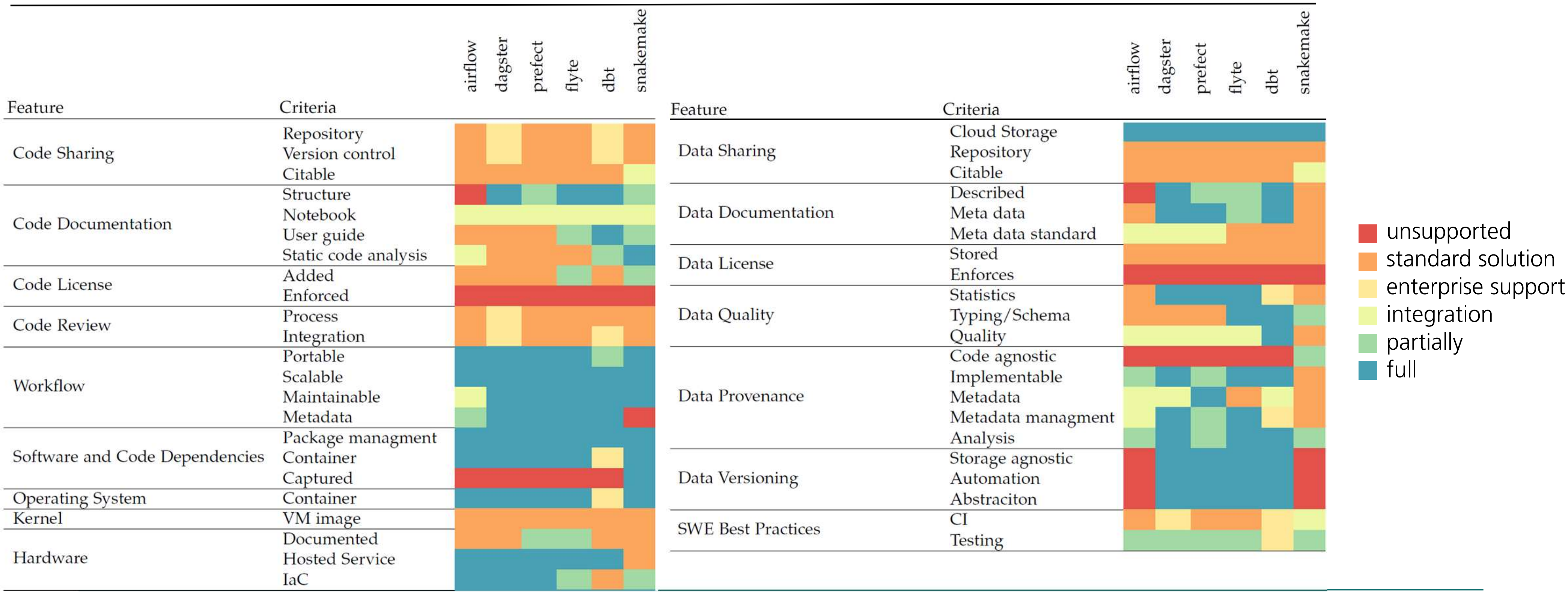
Selected tools for evaluation with respect to reproducible data preprocessing

Tool	Category	Description	Stars	Contributors
Airflow	Data pipelines	A platform to programmatically author, schedule, and monitor workflows.	30866	418
Prefect	Data pipelines	Prefect is a workflow orchestration tool empowering developers to build, observe, and react to data pipelines.	12278	170
Dagster	Data pipelines	An orchestration platform for the development, production, and observation of data assets.	7840	293
dbt	Data warehouse transformation workflows	dbt enables data analysts and engineers to transform their data using the same practices that software engineers use to build applications.	7230	256
Flyte	Data pipelines	Scalable and flexible workflow orchestration platform that seamlessly unifies data, ML and analytics stacks.	3561	121
Snakemake	Bioinformatic workflows	The Snakemake workflow management system is a tool to create reproducible and scalable data analyses.	1749	267

Results

Overview | Requirements for Reproducibility | Candidate Tool Landscape | Tool Selection | Tool Evaluation ▲

Reproducibility support for each criterion of a reproducibility feature with respect to the evaluated tools



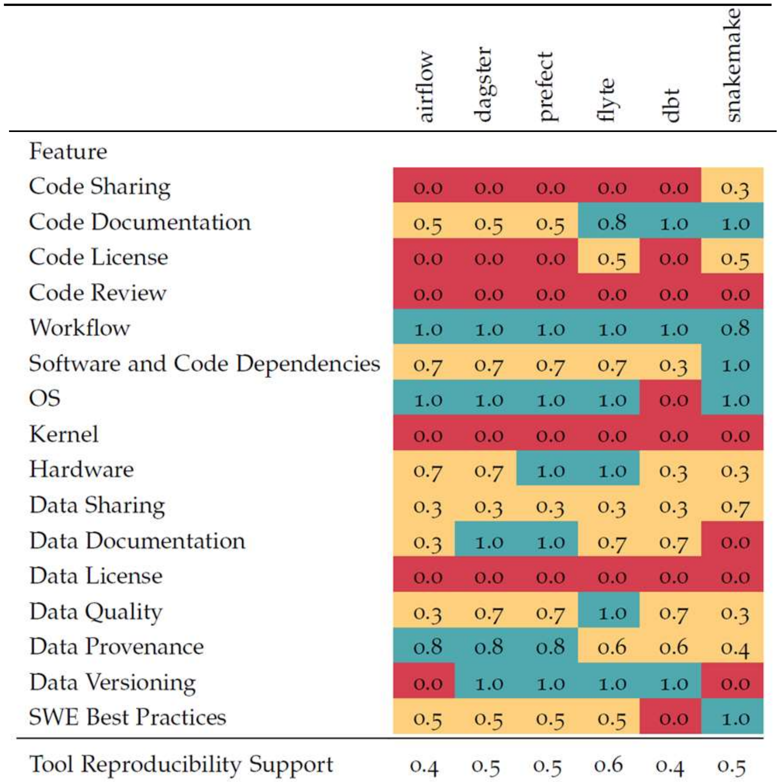
Feature	Criteria	airflow	dagster	prefect	flyte	dbt	snakemake
Code Sharing	Repository	🟧	🟨	🟧	🟧	🟨	🟧
	Version control	🟧	🟨	🟧	🟧	🟨	🟧
	Citable	🟧	🟧	🟧	🟧	🟧	▨
Code Documentation	Structure	🟥	🟦	🟩	🟦	🟦	🟩
	Notebook	▨	▨	▨	▨	▨	▨
	User guide	🟧	🟧	🟧	🟩	🟦	🟩
	Static code analysis	▨	🟧	🟧	🟧	🟩	🟦
Code License	Added	🟧	🟧	🟧	🟩	🟧	🟩
	Enforced	🟥	🟥	🟥	🟥	🟥	🟥
Code Review	Process	🟧	🟨	🟧	🟧	🟧	🟧
	Integration	🟧	🟨	🟧	🟧	🟨	🟧
Workflow	Portable	🟦	🟦	🟦	🟦	🟩	🟦
	Scalable	🟦	🟦	🟦	🟦	🟦	🟦
	Maintainable	▨	🟦	🟦	🟦	🟦	🟦
	Metadata	🟩	🟦	🟦	🟦	🟦	🟥
Software and Code Dependencies	Package managment	🟦	🟦	🟦	🟦	🟦	🟦
	Container	🟦	🟦	🟦	🟦	🟨	🟦
	Captured	🟥	🟥	🟥	🟥	🟥	🟦
Operating System	Container	🟦	🟦	🟦	🟦	🟨	🟦
Kernel	VM image	🟧	🟧	🟧	🟧	🟧	🟧
Hardware	Documented	🟧	🟧	🟩	🟩	🟧	🟧
	Hosted Service	🟦	🟦	🟦	🟦	🟦	🟧
	IaC	🟦	🟦	🟦	🟩	🟧	🟩

Feature	Criteria	airflow	dagster	prefect	flyte	dbt	snakemake
Data Sharing	Cloud Storage	🟦	🟦	🟦	🟦	🟦	🟦
	Repository	🟧	🟧	🟧	🟧	🟧	🟧
	Citable	🟧	🟧	🟧	🟧	🟧	▨
Data Documentation	Described	🟥	🟦	🟩	🟩	🟦	🟧
	Meta data	🟧	🟦	🟦	🟩	🟦	🟧
	Meta data standard	▨	▨	▨	🟧	🟧	🟧
Data License	Stored	🟧	🟧	🟧	🟧	🟧	🟧
	Enforces	🟥	🟥	🟥	🟥	🟥	🟥
Data Quality	Statistics	🟧	🟦	🟦	🟦	🟨	🟧
	Typing/Schema	🟧	🟧	🟧	🟦	🟦	🟩
	Quality	▨	▨	▨	▨	🟦	🟧
Data Provenance	Code agnostic	🟥	🟥	🟥	🟥	🟥	🟩
	Implementable	🟩	🟦	🟩	🟦	🟦	🟧
	Metadata	▨	▨	🟦	🟧	▨	🟧
	Metadata managment	▨	🟦	🟩	🟦	🟨	🟧
	Analysis	🟩	🟦	🟩	🟦	🟦	🟩
Data Versioning	Storage agnostic	🟥	🟦	🟦	🟦	🟦	🟥
	Automation	🟥	🟦	🟦	🟦	🟦	🟥
	Abstraciton	🟥	🟦	🟦	🟦	🟦	🟥
SWE Best Practices	CI	🟧	🟨	🟧	🟧	🟨	▨
	Testing	🟩	🟩	🟩	🟩	🟨	🟩

🟥 unsupported
🟧 standard solution
🟨 enterprise support
▨ integration
🟩 partially
🟦 full

Results

Reproducibility metric for each feature and tool



Feature	airflow	dagster	prefect	flyte	dbt	snakemake
Code Sharing	0.0	0.0	0.0	0.0	0.0	0.3
Code Documentation	0.5	0.5	0.5	0.8	1.0	1.0
Code License	0.0	0.0	0.0	0.5	0.0	0.5
Code Review	0.0	0.0	0.0	0.0	0.0	0.0
Workflow	1.0	1.0	1.0	1.0	1.0	0.8
Software and Code Dependencies	0.7	0.7	0.7	0.7	0.3	1.0
OS	1.0	1.0	1.0	1.0	0.0	1.0
Kernel	0.0	0.0	0.0	0.0	0.0	0.0
Hardware	0.7	0.7	1.0	1.0	0.3	0.3
Data Sharing	0.3	0.3	0.3	0.3	0.3	0.7
Data Documentation	0.3	1.0	1.0	0.7	0.7	0.0
Data License	0.0	0.0	0.0	0.0	0.0	0.0
Data Quality	0.3	0.7	0.7	1.0	0.7	0.3
Data Provenance	0.8	0.8	0.8	0.6	0.6	0.4
Data Versioning	0.0	1.0	1.0	1.0	1.0	0.0
SWE Best Practices	0.5	0.5	0.5	0.5	0.0	1.0
Tool Reproducibility Support	0.4	0.5	0.5	0.6	0.4	0.5

🟥 $0.0 \leq s_f < 0.3$

🟨 $0.3 \leq s_f \leq 0.7$

🟦 $0.7 < s_f \leq 1.0$

Summary and Outlook

- 16 features and respective criteria were proposed to help categorize the support for reproducible data preprocessing
- A candidate tool landscape in the AI and data domain was created to help identify relevant open-source tools in an iterative selection process
- A evaluation framework was designed, and six open-source tools were analyzed concerning their support for reproducible data preprocessing. None of them provides out-of-the-boc reproducibility.

Summary and Outlook

Summary Outlook

- Further formalize the reproducibility features and criteria to facilitate identifying the support level by a tool
- Describe the default solution for a specific feature to indicate that it has not to be reinvented by a tool
- Evaluate the available integrations
- Outline a tool stack, which could further help to support reproducible data preprocessing

Q&A

Simon Grimm

References

- [Fei+20] Melanie Feinberg, Will Sutherland, Sarah Beth Nelson, Mohammad Hossein Jarrahi, and Arcot Rajasekar. “The New Reality of Reproducibility: The Role of Data Work in Scientific Research.” In: Proceedings of the ACM on Human-Computer Interaction 4 (CSCW1 May 29, 2020)
- [Soh23] Emily Sohn. “The reproducibility issues that haunt health-careAI.” In: Nature 613.7943 (Jan. 9, 2023)
- [AKV19] Stamatios-Aggelos N. Alexandropoulos, Sotiris B. Kotsiantis, and Michael N. Vrahatis. “Data preprocessing in predictive data mining.” In: The Knowledge Engineering Review 34 (2019).
- [ATSO17] Mohammed Zuhair Al-Taie, Naomie Salim, and Adekunle Isiaka Obasa. “Successful Data Science Projects: Lessons Learned from Kaggle Competition.” In: Kurdistan Journal of Applied Research 2.3 (Aug. 27, 2017).
- [FAM+97] A. Famili,Wei-Min Shen, RichardWeber, and Evangelos Simoudis. “Data Preprocessing and Intelligent Data Analysis.” In: Intelligent Data Analysis 1.1 (Jan. 1, 1997)
- [GZ19] Carlos Vladimiro Gonzalez Zelaya. “Towards Explaining the Effects of Data Preprocessing on Machine Learning.” In: 2019 IEEE 35th International Conference on Data Engineering (ICDE). 2019
- [BR21] Sumon Biswas and Hridesh Rajan. “Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline.” In: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ESEC/FSE 2021.

References

- [IT18] Peter Ivie and Douglas Thain. “Reproducibility in Scientific Computing.” In: ACM Computing Surveys 51.3 (July 16, 2018)
- [Mö+17] Steffen Möller et al. “Robust Cross-Platform Workflows: How Technical and Scientific Communities Collaborate to Develop, Test and Share Best Practices for Data Analysis.” In: Data Science and Engineering 2.3 (Sept. 1, 2017)
- [Lei+21] Jeremy Leipzig, Daniel Nüst, Charles Tapley Hoyt, Karthik Ram, and Jane Greenberg. “The role of metadata in reproducible computational research.” In: Patterns 2.9 (Sept. 10, 2021).
- [HO20] Matthew Hartley and Tjelvar S. G. Olsson. “dtoolAI: Reproducibility for Deep Learning.” In: Patterns 1.5 (Aug. 14, 2020)
- [MJ21] Saúl Manzano and Adele C. M. Julier. “How FAIR are plant sciences in the twenty-first century? The pressing need for reproducibility in plant ecology and evolution.” In: Proceedings of the Royal Society B: Biological Sciences 288.1944 (Feb. 10, 2021).
- [SLKR21] Sheeba Samuel, Frank Löffler, and Birgitta König-Ries. “Machine Learning Pipelines: Provenance, Reproducibility and FAIRData Principles.” In: Provenance and Annotation of Data and Processes. Ed. by Boris Glavic, Vanessa Braganholo, and David Koop. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021
- [Wil+16] Mark D. Wilkinson et al. “The FAIR Guiding Principles for scientific data management and stewardship.” In: Scientific Data 3.1 (Mar. 15, 2016)

References

- [Bar+22] Michelle Barker et al. "Introducing the FAIR Principles for research software." In: Scientific Data 9.1 (Oct. 14, 2022).
- [PRSA18] Beatriz Pérez, Julio Rubio, and Carlos Sáenz-Adán. "A systematic review of provenance systems." In: Knowledge and Information Systems 57.3 (Dec. 1, 2018)
- [Per19] Jeffrey M. Perkel. "Workflow systems turn raw data into scientific knowledge." In: Nature 573.7772 (Sept. 2, 2019).
- [MG22] Beatriz M. A. Matsui and Denise H. Goya. "MLOps: Five Steps to Guide its Effective Implementation." In: 2022 IEEE/ACM 1st International Conference on AI Engineering – Software Engineering for AI (CAIN). 2022
- [Pim+21] João Felipe Pimentel, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. "Understanding and improving the quality and reproducibility of Jupyter notebooks." In: Empirical Software Engineering 26.4 (May 8, 2021)
- [Nü+20] Daniel Nüst, Vanessa Sochat, Ben Marwick, Stephen J. Eglen,Tim Head, Tony Hirst, and Benjamin D. Evans. "Ten simple rules for writing Dockerfiles for reproducible data science." In: PLOS Computational Biology 16.11 (Nov. 10, 2020).
- [Bar22